(54) Title: CODED DOMAIN ADAPTIVE LEVEL CONTROL OF COMPRESSED SPEECH
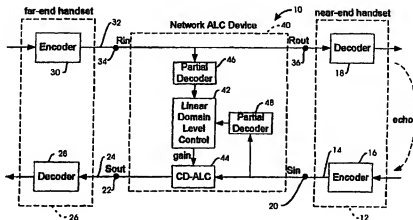
(57) Abstract: A communications system (10) transmits digital signals using a compression code comprising a predetermined plural-
ity of parameters including a first parameter. The parameters represent an audio signal comprising a plurality of audio characteristics
including a first characteristic related to the first parameter. The compression code is decodable by a plurality of decoding steps
including a first decoding step for decoding the parameters related to the first characteristic. A terminal (20) receives the digital
signals. A processor (40) responsive to the compression code reads at least the first parameter and generates at least a first parameter
value derived from the first parameter. The processor performs at least the first decoding step to generate decoded signals related to
the first characteristic of the audio signal. The processor also generates an adjusted first parameter value representing an adjustment
of the first characteristic, and derives an adjusted first parameter which is written into the compression code. Another terminal (22)
transmits the adjusted compression code. As a result, the delay required to adjust the first characteristic is reduced.

WO 01/03317 A1

TITLE OF THE INVENTION

CODED DOMAIN ADAPTIVE LEVEL CONTROL OF COMPRESSED SPEECH

CROSS-REFERENCE TO RELATED APPLICATIONS

This is a utility application corresponding to provisional application no. 60/142,136 entitled

"CODED DOMAIN ENHANCEMENT OF COMPRESSED SPEECH " filed July 2, 1999.

BACKGROUND OF THE INVENTION

The present invention relates to coded domain enhancement of compressed speech and in particular to coded domain adaptive level control and noise reduction in the coded domain.

Network enhancement of coded speech would normally require decoding, linear processing and re-encoding of the processed signal. Such a method is illustrated in Figure 1 and is very expensive. Moreover, the encoding process is often an order of magnitude more computationally intensive than the speech enhancement methods.

Speech compression is increasingly used in telecommunications, especially in cellular telephony and voice over packet networks. Past network speech enhancement techniques which operate in the linear domain have several shortcomings. For example, past network speech enhancement techniques which operate in the linear domain require decoding of compressed speech, performing the necessary enhancements and re-encoding of the speech. This processing can be computationally intensive, is especially prone to additional quantization noise, and can cause additional delay.

The maintenance of an optimum speech level is an important problem in the Public Switched Telephone Network (PSTN). Telephony customers expect a comfortable listening level to maximize comprehension of their conversation. The transmitted speech level from a telephone instrument depends on the speaker's volume and the position of the speaker relative to the microphone. If volume control is available on the telephone instrument, the listener could manually adjust it to a desirable level. However, for historical reasons, most telephone instruments do not have volume controls. Also, direct volume control by the listener does not address the need to maintain appropriate levels for network equipment. Furthermore, as technology is progressing towards the era of hands-free telephony especially in the case of mobile phones in vehicles, manual adjustment is considered cumbersome and potentially hazardous to the vehicle operators.

Maintaining speech quality has generally been the responsibility of network service providers; telephone instrument manufacturers typically have played a relatively minor role in meeting such responsibility. Traditionally, network service providers have provided tight specifications for equipment and networks with regard to speech levels. However, due to increased international voice traffic, deregulation, fierce competition and greater customer expectations, the network service providers have to ensure the proper speech levels with lesser influence over specifications and equipment used in other networks.

With the widespread introduction of new technology and protocols such as digital cellular telephony and voice over packet networks, the control of speech levels in the network has become more complex. One of the main reasons is the presence of

speech compression devices known as speech codecs (coder-decoder pairs) in the transmission path. Automatic level control (ALC) of speech signals becomes more difficult when speech codecs are present in the transmission path, while, in the linear domain, the digital speech samples are available for direct processing.

5   Figure 2 shows the network configuration of a linear domain ALC device 202. The ALC device processes the near-end speech signal (at port Sin). The far-end signal (at port Rin) is used for determining double-talk. ALC device 202 processes a digital near end speech signal in a typical transmission network and determines the gain required to attain a target speech level by measuring the current speech level.

10   Numerous algorithms can be devised to determine a suitable gain. For example, the ALC device could use a voice activity detector and apply new gain values only at the beginning of speech bursts. Furthermore, the maximum and minimum gain, and the maximum rate of change of the gain may all be constrained.   In general, ALC devices utilize (1) some form of power measurement scheme on the near end signal to

15   determine the current speech level, (2) a voice activity detector on the near end signal to demarcate speech bursts, and possibly (3) a double-talk detector on the far and near end signals to determine whether the near end signal contains echo.

           The ALC device determines the gain required to attain the target speech level by measuring the current speech level. Each digitized speech sample is multiplied by

20   a gain factor. The double-talk information is used to prevent adjusting the gain factor erroneously based on echo. Tellabs algorithms/products for level control include ALC, Sculptured Sound (SS) and the new TLC (Tellabs Level Control). These

-4-

algorithms are classified as linear domain algorithms since they operate directly on the linear/PCM signal.

The Tandem Free Operation (TFO) standard will be deployed in the Global System for Mobile Communications (GSM) digital cellular networks in the near future. The TFO standard applies to mobile-to-mobile calls. Under TFO, the speech signal is conveyed between mobiles in a compressed form after a brief negotiation period. The compressed speech is contained in TFO frames which bypass the transcoders in the network. This eliminates tandem voice codecs during mobile-to-mobile calls. The elimination of tandem codecs is known to improve speech quality in the case where the original signal is clean. Even in the case of clean speech, it may still be desirable to adjust the speech level to a suitable loudness level. Traditional methods for such level control would require decoding, processing and re-encoding the speech, which results in tandeming and is computationally-intensive. The coded domain approach avoids such tandeming and eliminates the need for full re-encoding. This document describes methods for speech level control in the coded domain. Specifically, level control in conjunction with the GSM FR and EFR coders is addressed.

BRIEF SUMMARY OF THE INVENTION

One preferred embodiment is useful in a communications system for transmitting digital signals using a compression code comprising a predetermined plurality of parameters including a first parameter, the parameters representing an audio signal comprising a plurality of audio characteristics including a first characteristic, the first parameter being related to the first characteristic, the

compression code being decodable by a plurality of decoding steps including a first

decoding step for decoding the parameters related to the first characteristic. In such

an environment, the first characteristic may be adjusted by reading at least the first

parameter in response to the digital signals. At least a first parameter value is derived

from the first parameter. An adjusted first parameter value representing an

adjustment of the first characteristic is generated in response to the digital signals and

the first parameter value. An adjusted first parameter is derived in response to the

adjusted first parameter value, and the first parameter of the compression code is

replaced with the adjusted first parameter. The preceding steps of reading, deriving,

generating and replacing preferably are performed by a processor. As a result of the

foregoing technique, the delay required to adjust the first characteristic   may be

reduced.

A second preferred embodiment is useful in a communication system for

transmitting digital signals comprising code samples comprising first bits using a

compression code and second bits using a linear code. The code samples represent an

audio signal have a plurality of audio characteristics, including a first characteristic.

In such an environment, the first characteristic may be adjusted without decoding the

compression code by adjusting the first bits and the second bits in response to the

second bits. The adjusting preferably is performed with a processor.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a schematic block diagram of a system for network enhancement of

coded speech in the linear domain.

-6-

Figure 2 is a schematic block diagram of a system for automatic level control (ALC).

Figure 3 is a schematic block diagram of a linear predictive coding (LPC) speech synthesis model.

Figure 4 is a schematic block diagram distinguishing coded domain digital speech parameters from linear domain digital speech samples.

Figure 5 is a schematic block diagram of a coded domain ALC system.

Figure 6 is a graph illustrating GSM full rate codec quantization levels for block maxima.

Figure 7a is a schematic block diagram of a backward adaptive standard deviation based quantizer.

Figure 7b is a schematic block diagram of a backward adaptive differential based quantizer.

Figure 8 is a schematic block diagram of an adaptive differential quantizer using a linear predictor.

Figure 9 is a schematic block diagram of a GSM enhanced full rate SLRP quantizer.

Figure 10 is a graph illustrating GSM enhanced full rate codec quantization levels for a gain correction factor.

Figure 11 is a schematic block diagram of one technique for performing ALC.

Figure 12 is a schematic block diagram of one technique for coded domain ALC.

Figure 13 is a flow diagram illustrating a technique for overflow/underflow prevention.

Figure 14 is a schematic block diagram of a preferred form of ALC system using feedback of the realized gain in ALC algorithms requiring past gain values.

Figure 15 is a schematic block diagram of one form of a coded domain ALC device.

Figure 16 is a schematic block diagram of a system for instantaneous scalar requantization for a GSM FR codec.

Figure 17 is a schematic block diagram of a system for differential scalar requantization for a GSM EFR codec.

Figure 18a is a graph showing a step in desired gain.

Figure 18b is a graph showing actual realized gain superimposed on the desired gain with a quantizer in the feedback loop.

Figure 18c is a graph showing actual realized gain superimposed on the desired gain resulting from placing a quantizer outside the feedback loop shown in Figure 19.

Figure 19 is a schematic block diagram of an ALC device showing a quantizer placed outside the feedback loop.

Figure 20 is a schematic block diagram of a simplified version of the ALC device shown in Figure 19.

Figure 21a is a schematic block diagram of a coded domain ALC implementation for ALC algorithms using feedback of past gain values with a quantizer in the feedback loop.

Figure 21b is a schematic block diagram of a coded domain ALC implementation for ALC algorithms using feedback of past gain values with a quantizer outside the feedback loop.

Figure 22 is a graph showing spacing between adjacent R_i values in an EFR codec, and more specifically showing EFR Codec SLRPs: $(R_{i+1} - R_i)$ against i.

Figure 23a is a diagram of a compressed speech frame of an EFR encoder illustrating the times at which various bits are received and the earliest possible decoding of samples as a buffer is filled from left to right.

Figure 23b is a diagram of a compressed speech frame of an FR encoder illustrating the times at which various bits are received and the earliest possible decoding of samples as a buffer is filled from left to right.

Figure 24 is a schematic block diagram of a preferred form of coded domain ALC system made in accordance with the invention.

Figure 25 is a schematic block diagram of a preferred form of SLRP quantization in GSM EFR.

Figure 26 is a schematic block diagram of an alternative form of SLRP quantization in GSM EFR.

Figure 27 is a schematic block diagram of a preferred form of re-encoding the SLRP in GSM EFR.

Figure 28 is a graph illustrating an exemplary speech signal.

Figure 29 is a graph illustrating exemplary speech level adjustment with CD-ALC for FR.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

While the invention will be described in connection with one or more embodiments, it will be understood that the invention is not limited to those embodiments. On the contrary, the invention includes all alternatives, modifications, and equivalents as may be included within the spirit and scope of the appended claims.

The following abbreviations are offered an aid to understanding the preferred embodiments:

ACELP	Algebraic Code Excited Linear Prediction
AE	Audio Enhancer
ALC	Adaptive or Automatic Level Control
CD	Compressed Domain or Coded Domain
CD-ALC	Coded Domain Adaptive Level Control
EFR	Enhanced Full Rate
ETSI	European Telecommunications Standards Institute
FR	Full Rate
GSM	Global System for Mobile Communications
ITU	International Telecommunications Union

LD-ALC	Linear Domain Adaptive Level Control (may be ALC, SS or TLC etc)
MR-ACELP	Multi-Rate ACELP
PCM	Pulse Code Modulation (ITU G.711)
RPE-LTP	Regular Pulse Excitation - Long Term Prediction
SLRP	Speech Level Related Parameter
SS	Sculptured Sound
TFO	Tandem Free Operation
TLC	Tellabs Level Control
VSELP	Vector Sum Excitation Linear Prediction

The following references are referred to in this specification :

[1] GSM 06.10, "Digital cellular telecommunication system (Phase 2); Full rate speech; Part 2: Transcoding", March 1998.

[2] GSM 06.60, "Digital cellular telecommunications system (Phase 2); Enhanced Full Rate (EFR) speech transcoding", June 1998.

[3] ITU-T Recommendation G.169 Draft 7, "Automatic Level Control Devices", July 1998.

In modern networks, speech signals are digitally sampled prior to transmission. Such digital (i.e. discrete-time discrete-valued) signals are referred to in this specification as being in the linear domain or in linear mode. The adjustment of the speech levels in such linear domain signals is accomplished by multiplying every sample of the signal by an appropriate gain factor to attain the desired target speech level.

Linear echo or acoustic echo may be present in the near end signal depending on the type of end path in the network. If such echo has significant power and is not

already cancelled by an echo canceller, then a double-talk detector may also be required. This is to ensure that the gain is not inadvertently increased due to the echo of the far end speech signal.

Digital speech signals that are typically carried in telephony networks usually undergo a basic form of compression such as pulse code modulation (PCM) before transmission. Such compression schemes are very inexpensive in terms of computations and delay. It is a relatively simple matter for the ALC device to convert the compressed digital samples to the linear domain, process the linear samples, and then compress the processed samples before transmission. As such, these signals can effectively be considered to be in the linear domain. In the context of this application, compressed, or coded speech will refer to speech that is compressed using advanced compression techniques that require significant computational complexity.

In this specification and claims, the terms linear code and compression code mean the following:

Linear code: By a linear code, we mean a compression technique that results in one coded parameter or coded sample for each sample of the audio signal. Examples of linear codes are PCM (A-law and μ-law) ADPCM (adaptive differential pulse code modulation), and delta modulation.

Compression code: By a compression code, we mean a technique that results in fewer than one coded parameter for each sample of the audio signal. Typically, compression codes result in a small set of coded parameters for each block or frame

-12-

of audio signal samples. Examples of compression codes are linear predictive coding based vocoders such as the GSM vocoders (HR, FR, EFR).

Speech compression, which falls under the category of lossy source coding, is commonly referred to as speech coding. Speech coding is performed to minimize the bandwidth necessary for speech transmission. This is especially important in wireless telephony where bandwidth is a scarce resource. In the relatively bandwidth abundant packet networks, speech coding is still important to minimize network delay and jitter. This is because speech communication, unlike data, is highly intolerant of delay. Hence a smaller packet size eases the transmission through a packet network. Several industry standard speech codecs (coder-decoder pairs) are listed in Table 1 for reference.

-13-

Table 1. Several Standardized Speech Codecs

Codec Name	Coding Method	Bit Rate (kbits/sec)	Standards Body
GSM Half Rate (HR)	VSELP	5.6	European Telecommunications Standards Institute
GSM Full Rate (FR)	RPE-LTP	13	
GSM Enhanced Full Rate (EFR)	ACELP	12.2	
GSM Adaptive Multi-Rate (AMR)	MR-ACELP	5.4 - 12.2	
ITU-T G.723.1	1. MP-MLQ 2. ACELP	6.3 5.3	International Telecommunications Union
ITU-T G.729	CS-ACELP	8	
ITU-T G.728	LD-CELP	16	

In speech coding, a set of consecutive digital speech samples is referred to as a speech frame. Given a speech frame, a speech encoder determines a small set of parameters for a speech synthesis model. With these speech parameters and the speech synthesis model, a speech frame can be reconstructed that appears and sounds very similar to the original speech frame. The reconstruction is performed by the speech decoder. It should be noted that, in most speech coders, the encoding process is much more computationally intensive than the decoding process. Furthermore, the MIPs required to attain good quality speech coding is very high. The processing capabilities of digital signal processing chipsets have advanced sufficiently only in recent years to enable the widespread use of speech coding in applications such as cellular telephone handsets.

The speech parameters determined by the speech encoder depend on the speech synthesis model used. For instance, the coders in Table 1 utilize linear predictive coding (LPC) models. A block diagram of a simplified view of the LPC speech synthesis model is shown in Figure 3. This model can be used to generate speech-like signals by specifying the model parameters appropriately. In this example speech synthesis model, the parameters include the time-varying filter coefficients, pitch periods, excitation vectors and gain factors. Basically, the excitation vector, $c(n)$, is first scaled by the gain factor, G. The result is then filtered by a pitch synthesis filter whose parameters include the pitch gain, g_p, and the pitch period, T, to obtain the total excitation vector, $u(n)$. This is then filtered by the LPC synthesis filter. Other models such as the multiband excitation model are also used in speech coding. In this context, it suffices to note that the speech parameters together with the assumed

model provide a means to remove the redundancies in the digital speech signal so as to achieve compression.

As shown in Figure 3, the overall DC gain is provided by G and ALC would primarily involve modifying G.

5   Among the speech parameters that are generated each frame by a typical speech encoder, some parameters are concerned with the spectral and/or waveform shapes of the speech signal for that frame. These parameters typically include the LPC coefficients and the pitch information in the case of the LPC speech synthesis model. In addition to these parameters that provide spectral information, there are usually

10   parameters that are directly related to the power or energy of the speech frame. These speech level related parameters (SLRPs) are the key to performing ALC of coded speech. Several examples of such SLRPs will be provided below.

The first three GSM codecs in Table 1 will now be discussed. All of the first three coders process speech sampled at 8kHz and assume that the samples are

15   obtained as 13-bit linear PCM values. The frame length is 160 samples (20ms). Furthermore, they divide each frame into four subframes of 40 samples each. The SLRPs for these codecs are listed in Table 2.

Table 2. Speech Level Related Parameters in GSM Speech Codecs

Codec Name	SLRP	Description
GSM Half Rate	$R(0)$	$R(0)$ is the average signal power of the speech frame. The signal power is computed using an analysis window which is centered over the last 100 samples of the frame. The signal power in decibels is quantized to 32 levels which are spaced uniformly in 2dB steps.
GSM Full Rate	x_{max}	x_{max} is the maximum absolute value of the elements in the subframe

		excitation vector. x_{max} is also termed the block maximum. All the other subframe excitation elements are normalized and then quantized with respect to this maximum. The maximum is quantized to 64 levels non-uniformly.
GSM Enhanced Full Rate	γ_{gc}	γ_{gc} is the gain correction factor between a gain factor, g_c, used to scale the subframe excitation vector and a gain factor, g'_c, that is predicted using a moving average model, i.e. $\gamma_{gc} = g_c / g'_c$. The correction factor is quantized to 32 levels non-uniformly.

Depending on coder, the SLRP may be specified each subframe (e.g. the GSM FR and EFR codecs) or once per frame (e.g. the GSM HR codec).

Throughout this specification, the same variable with and without a caret above it will be used to denote the unquantized and quantized values that it holds, e.g. γ_{gc} and $\hat{\gamma}_{gc}$ are the unquantized and quantized gain correction factors in the GSM EFR standard. Note that only the quantized SLRP, $\hat{\gamma}_{gc}$, will be available at the ALC device.

The quantized and corresponding unquantized parameters are related through the quantization function, Q(.), e.g. $\hat{\gamma}_{gc}$=Q(γ_{gc}). We use the notation somewhat liberally to include not just this transformation but, depending on the context, the determination of the index of the quantized value using a look-up table or formula.

The quantization function is a many-to-one transformation and is not invertible. However, we use the 'inverse' quantization function, $Q^{-1}(.)$, to denote the conversion of a given index to it corresponding quantized value using the appropriate look-up table or formula.

Turning now to Figure 4, that Figure distinguishes the coded domain from the linear domain. In the linear domain, the digital speech samples are directly available for processing. The coded domain refers to the output of speech encoders or the input of the speech decoders, which should be identical if there are no channel errors. In this context, the coded domain includes both the speech parameters and the methods used to quantize or dequantize these parameters. The speech parameters that are determined by the encoder undergo a quantization process prior to transmission. This quantization is critical to achieving bit rates lower than that required by the original digital speech signal. The quantization process often involves the use of look-up tables. Furthermore, different speech parameters may be quantized using different techniques.

Processing of speech in the coded domain involves directly modifying the quantized speech parameters to a different set of quantized values allowed by the quantizer for each of the parameters. In the case of ALC, the parameters being modified are the SLRPs. The coded domain counterpart to the linear domain ALC configuration of Figure 2 is shown in Figure 5. Note that the codecs used for the two directions of transmission shown may not be identical. Furthermore, the codecs used may change over time. Hence the coded domain ALC algorithm preferably operates robustly under such changing conditions.

The quantization of a single speech parameter is termed scalar quantization. When a set of parameters are quantized together, the process is called vector quantization. Vector quantization is usually applied to a set of parameters that are related to each other in some way such as the LPC coefficients. Scalar quantization is

-18-

generally applied to a parameter that is relatively independent of the other parameters. A mixture of both types of quantization methods is also possible. As the SLRPs are usually scalar quantized, focus is placed on the most commonly used scalar quantization techniques.

5   When a parameter is quantized instantaneously, the quantization process is independent of the past and future values of the parameter. Only the current value of the parameter is used in the quantization process. The parameter to be quantized is compared to a set of permitted quantization levels. The quantization level that best matches the given parameter in terms of some closeness measure is chosen to

10   represent that parameter. Usually, the permitted quantization levels are stored in a look-up table at both the encoder and the decoder. The index into the table of the chosen quantization level is transmitted by the encoder to the decoder. Alternatively, given an index, the quantization level may be determined using a mathematical formula. The quantization levels are usually spaced non-uniformly in the case of

15   SLRPs. For instance, the block maxima, x_{max}, in the GSM FR codec which has a range [0,32767] is quantized to the 64 levels shown in Figure 6. In this quantization scheme, the level that is closest but higher than x_{max} is chosen. Note that the vertical axis which represents the quantization levels is plotted on a logarithmic scale.

Instantaneous quantization schemes suffer from higher quantization errors due

20   to the use of a fixed dynamic range. Thus, adaptive quantizers are often used in speech coding to minimize the quantization error at the cost of greater computational complexity. Adaptive quantizers may utilize forward adaptation or backward adaptation. In forward adaptation schemes, extra side information regarding the

dynamic range has to be transmitted periodically to the decoder in addition to the quantization table index. Thus, such schemes are usually not used in speech coders. Backward adaptive quantizers are preferred because they do not require transmission of any side information. Two general types of backward adaptive quantizers are commonly used: standard deviation based and differential. These are depicted in Figure 7.

In the standard deviation based quantization scheme of Figure 7(a), the standard deviation of previous parameter values are used to determine a normalization factor for the current parameter value, $\gamma(n)$. The normalization factor divides $\gamma(n)$ prior to quantization. This normalization procedure allows the quantization function, Q(.), to be designed for unit variance. The look-up table index of the normalized and quantized value, $\hat{\gamma}_{norm}(n)$, is transmitted to the dequantizer where the inverse process is performed. In order for the normalization and denormalization processes to be compatible, a quantized version of the normalization factor is used at both the quantizer and dequantizer. In some variations of this scheme, decisions to expand or compress the quantization intervals may be based simply on the previous parameter input only.

In the backward adaptive differential quantization scheme of Figure 7(b), the correlation between current and previous parameter values is used to advantage. When the correlation is high, a significant reduction in the quantization dynamic range can be achieved by quantizing the prediction error, $r(n)$. The prediction error is the difference between the actual and predicted parameter values. The same predictor

-20-

for $\gamma(n)$ must be used at both the quantizer and the dequantizer. A linear predictor, P(z), which has the following form is usually used:

$$P(z) = \sum_{k=1}^{p} b'_k z^{-k} \qquad (1)$$

It can be shown readily that the differential quantization scheme can also be represented as in Figure 7 when a linear predictor, P(z), is used. Note that if we approximate the transfer function P(z)/[1-P(z)] by the linear predictor, $P_1(z) = \sum_{k=1}^{p} b_k z^{-k}$, then a simpler implementation can be achieved. This simpler differential technique is used in the GSM EFR codec for the quantization of a function of the gain correction factor, γ_{gc}. In this codec, a fourth order linear predictor with fixed coefficients, [b1,b2,b3,b4] = [0.68, 0.58, 0.34, 0.19], is used at both the encoder and the decoder.

In the EFR codec, $g_c(n)$ denotes the gain factor that is used to scale the excitation vector at subframe n. This gain factor determines the overall signal level. The quantization of this parameter utilizes the scheme shown in Figure 8 but is rather indirect. The actual 'gain' parameter that is transmitted is actually a correction factor between $g_c(n)$ and the predicted gain, $g_c'(n)$. The correction factor, $\gamma_{gc}(n)$, defined as

$$\gamma_{gc}(n) = \frac{g_c(n)}{g_c'(n)} \qquad (2)$$

is considered the actual SLRP because it is the only parameter related to the overall speech level that is accessible directly in the coded domain.

At the encoder, once the best $g_c(n)$ for the current subframe n is determined, it is divided by the predicted gain to obtain $\gamma_{gc}(n)$. The predicted gain is given by

$$g_c^1(n) = 10^{0.05[\tilde{E}(n)-E_1(n)+\overline{E}]} \qquad (3)$$

A 32-level non-uniform quantization is performed on $\gamma_{gc}(n)$ to obtain $\hat{\gamma}_{gc}(n)$. The corresponding look-up table index is transmitted to the decoder. In equation (3), $\overline{E}$ is a constant, $E_1(n)$ depends only on the subframe excitation vector, and $\tilde{E}(n)$ depends only on the previously quantized correction factors. The decoder, thus, can obtain the predicted gain in the same manner as the encoder using (3) once the current subframe excitation vector is received. On receipt of the correction factor $\hat{\gamma}_{gc}(n)$, the quantized gain factor can be computed as $\hat{g}_c(n) = \hat{\gamma}_{gc}(n)g_c^1(n)$ using the definition in equation (2).

The quantization of the SLRP, γ_{gc}, is illustrated in Figure 9. In this Figure, $R(n)$ denotes the prediction error given by

$$R(n) = E(n) - \tilde{E}(n) = 20 \log \gamma_{gc}(n) \qquad (4)$$

Note that the actual information transmitted from the encoder to the decoder are the bits representing the look-up table index of the quantized $R(n)$ parameter, $\hat{R}(n)$. This detail is omitted in Figure 9 for simplicity. Since the preferred ALC technique does not affect the channel bit error rate, it is assumed that the transmitted and received parameters are identical. This assumption is valid because the result of

undetected or uncorrected errors will result in noisier decoded speech regardless of whether ALC is performed.

The quantization of the SLRP at the encoder is performed indirectly by using the mean-removed excitation vector energy each subframe. $E(n)$ denotes the mean-removed excitation vector energy (in dB) at subframe n and is given by

$$E(n) = 10 \, \log\left(\frac{1}{N} g_c^2 \sum{}_{i=0}^{N-1} c^2(n) \right) - \overline{E}$$
$$= 20 \log g_c + 10 \, \log\left(\frac{1}{N} \sum{}_{i=0}^{N-1} c^2(n) \right) - \overline{E} \qquad (5)$$

Here N = 40 is the subframe length and $\overline{E}$ is constant. The middle term in the second line of equation (5) is the mean excitation vector energy, $E_i(n)$, i.e.

$$E_i(n) = 10 \, \log\left(\frac{1}{N} \sum{}_{i=0}^{N-1} c^2(i) \right) \qquad (6)$$

The excitation vector $\{c(i)\}$ is decoded at the decoder prior to the determination of the SLRP. Note that the decoding of the excitation vector is independent of the decoding of the SLRP. It is seen that E(n) is a function of the gain factor, g_c. The quantization of $\gamma_{gc}(n)$ to $\hat{\gamma}_{gc}(n)$ indirectly causes the quantization of g_c to $\hat{g}_c$. This quantized gain factor is used to scale the excitation vector, hence setting the overall level of the signal synthesized at the decoder. $\tilde{E}(n)$ is the predicted energy given by

-23-

$$\tilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i) \qquad (7)$$

where $\left\{\hat{R}(n-i)\right\}$ are previously quantized values.

The preferred method of decoding the gain factor, g_c, will now be discussed. First, the decoder decodes the excitation vector and computes $E_I(n)$ using equation (6). Second, the predicted energy is computed using previously decoded gain correction factors using equation (7). Then the predicted gain, $g'(c)$, is computed using equation (3). Next, the received index of the correction factor for the current subframe is used to obtain $\hat{\gamma}_{gc}(n)$ from the look-up table. Finally, the quantized gain factor is obtained as $\hat{g}_c(n) = \hat{\gamma}_{gc}(n) g'_c(n)$. The 32 quantization levels for $\hat{\gamma}_{gc}(n)$ are illustrated in Figure 10. Note that the vertical axis in Figure 10 which represents the quantization levels is plotted on a logarithmic scale.

Regardless of the particular codec used, several general approaches to performing ALC in the coded domain may be devised. Figure 5 illustrated a preferred location of an ALC device operating on coded speech. With reference to this Figure, possible implementations of the ALC device will be discussed.

The most straightforward method for performing ALC is shown in Figure 11. The coded speech is decoded to the linear domain, ALC is performed on the linear domain signal in the usual manner, and then the linear speech is re-encoded. As discussed above, such a technique is extremely expensive in terms of MIPs, processing and buffering delay. Note that the encoding process is usually an order of magnitude more expensive than the decoding process. The encoding process also adds

quantization noise that can be observed in the decoded signal. Since there are two encoder-decoder pairs placed in tandem in this approach, the quantization noise is approximately doubled (when the ALC device gain is unity). This results in an undesirable degradation in speech quality.

Since the SLRP determines the speech level, it would be highly beneficial, to devise ALC techniques that only modify the SLRP. This would avoid the computational complexity and quality degradation associated with total re-encoding of the level-modified speech signal. A novel coded domain ALC approach that modifies only the SLRP is illustrated in Figure 12. Note that the details of the ALC algorithm will depend on the particular codec used. However, the approach described here is applicable in general to any codec.

In this approach, the quantized SLRP is decoded (e.g., read) from the coded domain signal (e.g., compression code signal) and multiplied (e.g., adjusted) by a gain factor determined by the ALC algorithm. (After multiplication, the SLRP may be considered an adjusted SLRP value.) The result is then requantized (e.g., to form an adjusted SLRP). The coded domain signal is appropriately modified to reflect the change in the SLRP. (For example, the adjusted SLRP may be substituted for the original SLRP.) For instance, any form of error protection used on the coded domain signal must be appropriately reinstated. The ALC device may require measures of the speech level, voice activity and double-talk activity to determine the gain that is to be applied to the SLRP. This may require the decoding of the coded domain signal to some extent.

-25-

For most codecs, only a partial decoding of the coded speech is necessary to perform ALC. The speech is decoded to the extent necessary to extract (e.g., read) the SLRP as well as other parameters essential for obtaining sufficiently accurate speech level, voice activity and double-talk measurements. Some examples of situations where only partial decoding suffices include:

1) In CELP decoders, a post-filtering process (i.e., decoding step) is performed on the signal decoded using the LPC-based model. This post-filtering helps to reduce quantization noise but does not change the overall power level of the signal. Thus, in partial decoding of CELP-coded speech, the post-filtering process (i.e., decoding step) can be avoided for economy.

2) Some form of silence suppression scheme is often used in cellular telephony and voice over packet networks. In these schemes, coded speech frames are transmitted only during voice activity and very little transmission is performed during silence. The decoders automatically insert some comfort noise during the silence periods to mimic the background noise from the other end. One example of such a scheme used in GSM cellular networks is called discontinuous transmission (DTX). By monitoring the side information that indicates silence suppression, the decoder in the ALC device can completely avoid decoding the signal during silence. In such cases, the determination of voice and double-talk activities can also be simplified in the ALC device.

3) In the proposed Tandem-Free Operation (TFO) standard for speech codecs in GSM networks, the coded speech bits for each channel will be carried through the

wireline network between base stations at 64 kbits/sec. This bitstream can be divided into 8-bit samples. The 2 least significant bits of each sample will contain the coded speech bits while the upper 6 bits will contain the bits corresponding to the appropriate PCM samples. The conversion of the PCM information to linear speech is

5   very inexpensive and provides a somewhat noisy version of the linear speech signal. It is possible to use this noisy linear domain speech signal to perform the necessary voice activity, double-talk and speech level measurements as is usually done in linear domain ALC algorithms. Thus, in this case, only a minimal amount of decoding of the coded domain speech parameters is necessary. The SLRP and any other parameters

10   that are required for the requantization of the SLRP would have to be decoded. The other parameters would be decoded only to the extent necessary for requantization of the SLRP. This will be clear from the examples that will follow in later sections.

Thus, we see that it is possible to implement an ALC device that only performs partial decoding and re-encoding, hence minimizing complexity and

15   reducing quantization noise. However, the ALC approach illustrated in Figure 12 is sub-optimal and may require improvement. The sub-optimality is due to the implicit assumption that the process of gain determination is independent of SLRP requantization. In general, this assumption may not be valid.

There are three main reasons for the possible sub-optimality of the method of

20   Figure 12. They are listed below. First, note that requantization results in a realized SLRP that usually differs from the desired value. Hence the desired gain that was applied by the Gain Determination block will differ from the gain that will be realized when the signal is decoded. When decoding, overflow or underflow problems may

arise due to this difference because the speech signal may be over-amplified or over-suppressed, respectively. Second, some ALC algorithms may utilize the past desired gain values to determine current and future desired gain values. Since the desired gain values do not reflect the actual realized gain values, such algorithms may perform

5   erroneously when applied as shown in Figure 12. Third, the requantization process can sometimes result in undesirable reverberations in the SLRP. This can cause the speech level to be modulated unintentionally, resulting in a distorted speech signal. Such SLRP reverberations are encountered in feedback quantization schemes such as differential quantization.

10   Turning now to Figure 13, to overcome the overflow/underflow problems, the iterative scheme of Figure 13 can be incorporated in the Gain Determination block. Basically, after deciding on a desired gain value, the realized gain value after requantization of the SLRP may be computed. The realized gain is checked to see if overflow or underflow problems could occur. This could be accomplished, for

15   example, by determining what the new speech level would be by multiplying the realized gain by the original speech level. Alternatively, a speech decoder could be used in the ALC device to see whether overflow/underflow actually occurs. Either way, if the realized gain value is deemed to be too high or too low, the new SLRP is reduced or increased, respectively, until the danger of overflow/underflow is

20   considered to be no longer present.

In ALC algorithms where past desired gain values are fed back into the algorithm to determine current and future gain values, the following modification must be made. Basically, the gain that is fed back should be the realized gain after the

-28-

SLRP requantization process, not the desired gain. A preferred approach is shown in Figure 14. If the desired gain was used in the feedback loop instead of the realized gain, the controller would not be tracking the actual decoded speech signal level, resulting in erroneous level control.

Note that the iterative scheme for overflow/underflow prevention of Figure 13 may also be incorporated into the Gain Determination block of Figure 14.

Finally, the methods to avoid SLRP reverberations in feedback-based quantization schemes will be discussed in detail below. In general, these methods preferably include the integration of the gain determination and SLRP requantization techniques.

Hence the joint design and implementation of the Gain Determination block and SLRP Requantization block is preferred to prevent overflow and underflow problems during decoding, ensure proper tracking by feedback-based ALC systems, and avoid the oscillatory effects introduced by feedback quantization schemes. Figure 15 illustrates the general configuration of an ALC device that uses joint gain determination and SLRP requantization. The details will depend on the particular ALC device.

The techniques for requantization of SLRPs will now be discussed. In most speech encoders, the quantization of the SLRP is performed using either instantaneous scalar quantization or differential scalar quantization, which were discussed above. The requantization of the SLRPs for these particular cases will be described while noting that the approaches may be easily extended to any other quantization scheme.

The joint determination of the gain and SLRP requantization in the ALC device configuration of Figure 15 may utilize the requantization techniques described here.

The original value of the quantized SLRP will be denoted by $\hat{\gamma}(n)$, where n is the frame or subframe index. The set of m quantization table values will be denoted by $\{\hat{\gamma}_1 ... \hat{\gamma}_m\}$. Depending on the speech coder, these values may, instead, be defined using a mathematical formula. The desired gain determined by the ALC device will be denoted by $g(n)$. The realized gain after SLRP requantization will be denoted by $\hat{g}(n)$. In instantaneous scalar requantization, the goal is to minimize the difference between $g(n)$ and $\hat{g}(n)$. The basic approach involves the selection of the quantization table index, k, as

$$k = \arg\min_i \|g(n)\hat{\gamma}(n) - \hat{\gamma}_i\| \qquad (8)$$

The requantized SLRP is then given by $\hat{\gamma}_{alc}(n) = \hat{\gamma}_k$.

If overflow and underflow prevention are desired, then the iterative scheme described in Figure 13 may be used. In another approach for overflow/underflow prevention, the partial decoding of the speech samples using the requantized SLRP may be performed to the extent necessary. This, of course, involves additional complexity in the algorithm. The decoded samples can then be directly inspected to ensure that overflow or underflow has not taken place.

Note that for a given received $\hat{\gamma}(n)$, there are m possible realized gain values. For each quantization table value, all the realized gains can be precomputed and stored. This would require the storage of m^2 realized gain values, which is often

feasible since m is usually a small power of two, e.g. $m = 32$ in the GSM EFR codec and $m = 64$ in the GSM FR codec.

If the SLRP quantization table values are uniformly spaced (either linearly or logarithmically), then it is possible to simplify the scalar requantization process. This simplification is achieved by allowing only a discrete set of desired gain values in the ALC device. These desired gain values preferably have the same spacing as the SLRP quantization values, with 0dB being one of the gains. This ensures that the desired and realized gain values will always be aligned so that equation (8) would not have to be evaluated for each table value. Hence the requantization is greatly simplified. The original quantization index of the SLRP is simply increased or decreased by a value corresponding to the desired gain value divided by the SLRP quantization table spacing. For instance, suppose that the SLRP quantization table spacing is denoted by Δ. Then the discrete set of permitted desired gain values would be $1 + \{..., -2\Delta, -\Delta, 0, \Delta, 2\Delta, ...\}$ if the SLRP quantization table values are uniformly spaced linearly, and $0 + \{..., -2\Delta, -\Delta, 0, \Delta, 2\Delta, ...\}$ if the SLRP quantization table values are uniformly spaced logarithmically. If the desired gain value was $1 + k_1\Delta$ (linear case) or $k_1\Delta$ (logarithmic case), then the index of the requantized SLRP is simply obtained by adding k_1 to the original quantization index of the SLRP.

Note that this low complexity instantaneous scalar requantization technique can be applied even if the SLRP quantization table values are not uniformly spaced. In this case, Δ would be the average spacing between adjacent quantization table values, where the average is performed appropriately using either linear or logarithmic distances between the values.

An example of instantaneous scalar requantization is shown for the GSM FR codec in Figure 16. This codec's SLRP is the block maximum, x_{max}, which is transmitted every subframe. The Q and Q^{-1} blocks represent the SLRP requantization and dequantization, respectively. The index of the block maximum is first dequantized using the look-up table to obtain x_{max}. Then, x_{max} is multiplied by the desired gain to obtain $x_{max,ALC}$ which is then requantized by using the look-up table. The index of the requantized x_{max} is then substituted for the original value in the bitstream before being sent out. This requantization technique forms the basic component of all the schemes described in Figures 12-15 when implementing coded domain ALC for the GSM FR standard.

Novel techniques for differential scalar requantization will now be discussed. The GSM EFR codec will be used as an example for illustrating the implementation of coded domain ALC using this requantization technique.

Figure 17 shows a general coded domain ALC technique with only the components relevant to ALC being shown. Note that (G(n) denotes the original *logarithmic* gain value determined by the encoder. In the case of the EFR codec, G(n) is equal to E(n) defined in equation (5) and R(n) is as defined in equation (4). The ALC device determines the desired gain, $\Delta G(n)$. The SLRP, $\hat{R}(n)$, is modified by the ALC device to $\hat{R}_{ALC}(n)$ based on the desired gain. The realized gain, $\Delta \hat{R}(n)$, is the difference between original and modified SLRPs, i.e.

$$\Delta \hat{R}(n) = \hat{R}_{ALC}(n) - \hat{R}(n) \qquad (9)$$

-32-

Note that this is different from the actual gain realized at the decoder which, under steady-state conditions, is $[1 + P_1(1)]\Delta\hat{R}(n)$. To make the distinction clear, we will refer to the former as the *SLRP realized gain* and the latter as the *actual realized gain*. The actual realized gain is essentially an amplified version of the SLRP realized gain due to the decoding process, under steady-state conditions. By steady-state, it is meant that $\Delta G(n)$ is kept constant for a period of time that is sufficiently long so that $\Delta\hat{R}(n)$ is either steady or oscillates in a regular manner about a particular level.

This method for differential scalar requantization basically attempts to mimic the operation of the encoder at the ALC device. If the presence of the quantizers at the encoder and the ALC device is ignored, then both the encoder and the ALC device would be linear systems with the same transfer function, $1/[1 + P_1(z)]$, with the result that $\hat{G}_{ALC}(n) = G(n) + \Delta G(n)$. However, due to the quantizers which make these systems non-linear, this relationship is only approximate. Hence, the decoded gain given by

$$\hat{G}_{ALC}(n) = G(n) + \Delta G(n) + \text{quantization error}$$

(10)

where $(\Delta G(n) + \text{quantization error})$ is the actual realized gain.

The feedback of the SLRP realized gain, $\Delta\hat{R}(n)$, in the ALC device can cause undesirable oscillatory effects. As an example, we will demonstrate these oscillatory effects when the GSM EFR codec is used. Recall that, for this codec, $P_1(z)$ has four

-33-

delays elements. Each element could contain one of 32 possible values. Hence the non-linear system in the ALC device can be in any one of over a million possible states at any given time. This is mentioned because the behavior of this non-linear system is heavily influenced by its initial conditions.

The reverberations in the actual realized gain in response to a step in the desired gain, $\Delta G(n)$, will now be illustrated. For simplicity, it is assumed that the original SLRP, $\hat{R}(n)$, is constant over 100 subframes, and that the memory of $P_l(z)$ is initially zero. Figure 18(a) shows the step in the desired gain. Figure 18(b) shows the actual realized gain superimposed on the desired gain. Although the initial conditions and the original SLRP will determine the exact behavior, the reverberations in the actual realized gain shown here are quite typical.

The reverberations in the SLRP realized gain shown in Figure 18(b) cause a modulation of the speech signal and can result in audible distortions. Thus, depending on the ALC specifications, such reverberations may be undesirable. The reverberations can be eliminated by 'moving' the quantizer outside the feedback loop as shown in Figure 19. (In this embodiment, the computation of $\Delta\hat{R}(n)$ is unnecessary but is included for comparison to Figure 17.)

Placing the quantizer outside the feedback loop results in the actual realized gain shown in Figure 18(c), superimposed on the desired gain. It should be noted that, although reverberations are eliminated, the average error (i.e. the average difference between the desired and actual realized gains) is higher than that shown in Figure 18(b). Specifically, in these examples, the average error during steady state operation

-34-

of the requantizer with and without the quantizer in the feedback loop are 0.39dB and 1.03dB, respectively.

The ALC apparatus of Figure 19 can be simplified as shown in Figure 20, resulting in savings in computation. This is done by replacing the linear system 5 $1/[1 + P_1(z)]$ with the constant, $\dfrac{1}{[1 + P_1(1)]}$.

For the purposes of ALC, this simpler implementation is often found to be satisfactory especially when the desired gains are changed relatively infrequently. By infrequent changes, it is meant that the average number of subframes between changes is much greater than the order of $P_1(z)$.

10   Some ALC algorithms may utilize past gain values to determine current and future gain values. In such feedback-based ALC algorithms, the gain that is fed back should be the actual realized gain after the SLRP requantization process, not the desired gain. This was discussed above in conjunction with Figure 14.

Differential scalar requantization for such feedback-based ALC algorithms can 15   be implemented as shown in Figure 21. In these implementations, the ALC device is mimicking the actions of the decoder to determine the actual realized gain.

If a simplified ALC device implementation similar to Figure 19 is desired in Figure 21(b), then the linear system $\dfrac{1}{[1 + P_1(z)]}$ may be replaced with the constant multiplier, $\dfrac{1}{[1 + P_1(1)]}$. A further simplification can be achieved in Figure 21(b) by

-35-

replacing the linear system $1 + P_1(z)$ with the constant multiplier $1 + P_1(1)$, although accuracy in the calculation of the actual realized gain is somewhat reduced. In a similar manner, the implementation shown in Figure 21(a) can be simplified by replacing the linear system by with the constant multiplier $P_1(1)$.

5   In applications that are tolerant to reverberations but require higher accuracy in matching the desired and actual realized gains, any of the methods described earlier that have quantizers within the feedback loop may be used. For applications that cannot allow reverberations in the actual realized gains but can tolerate lower accuracy in matching the desired and actual realized gains, any of the methods

10   described earlier that have quantizers outside the feedback loop may be used.

         Large buffering, processing and transmission delays are already incurred by speech coders. Further processing of the coded speech for speech enhancement purposes can add additional delay. Such additional delay is undesirable as it can potentially make telephone conversations less natural. Furthermore, additional delay

15   may reduce the effectiveness of echo cancellation at the handsets, or alternatively, increase the necessary complexity of the echo cancellers for a given level of performance. It should be noted that implementation of ALC in the linear domain will always add at least a frame of delay due to the buffering and processing requirements for decoding and re-encoding. For the codecs listed in Table 1, note that each frame is

20   20ms long. However, coded domain ALC can be performed with a buffering delay much less than one frame.

The EFR encoder compresses a 20ms speech frame into 244 bits. At the decoder in the ALC device, the earliest point at which the first sample can be decoded is after the reception of bit 91 as shown in Figure 23(a). This represents a buffering delay of approximately 7.46ms. It turns out that sufficient information is received to decode not just the first sample but the entire first subframe at this point. Similarly, the entire first subframe can be decoded after about 7.11ms of buffering delay in the FR decoder.

The remaining subframes, for both coders, require shorter waiting times prior to decoding. Note that each subframe has an associated SLRP in both the EFR and FR coding schemes. This is generally true for most other codecs where the encoder operates at a subframe level.

From the above, it can be realized that ALC in the coded domain can be performed subframe-by-subframe rather than frame-by-frame. As soon as a subframe is decoded and the necessary level measurements are updated, the new SLRP computed by the ALC device can replace the original SLRP in the received bitstream.

The delay incurred before the SLRP can be decoded is determined by the position of the bits corresponding to the SLRP in the received bitstream. In the case of the FR and EFR codecs, the position of the SLRP bits for the first subframe determines this delay.

Most ALC algorithms determine the gain for a speech sample only after receiving that sample. This allows the ALC algorithm to ensure that the speech signal does not get clipped due to too large a gain, or underflow due to very low gains.

-37-

However, in a robust ALC algorithm, both overflow and underflow are events that have low likelihoods. As such, one can actually determine gains for samples based on information derived only from previous samples. This concept is used to achieve near-zero buffering delay in coded domain ALC for some speech codecs.

5   Basically, the ALC algorithm must be designed to determine the gain for the current subframe based on previous subframes only. In this way, almost no buffering delay will be necessary to modify the SLRP. As soon as the bits corresponding to the SLRP in a given subframe are received, they will first be decoded. Then the new SLRP will be computed based on the original SLRP and information from the previous subframes only. The original SLRP bits will be replaced with the new SLRP bits. There is no need to wait until all the bits necessary to decode the current subframe are received. Hence, the buffering delay incurred by the algorithm will depend on the processing delay which is small. Information about the speech level is derived from the current subframe only after replacement of the SLRP for the current subframe.

Note that most ALC algorithms can be easily converted to operate in this delayed fashion. Although there is a small risk of overflow or underflow, such risk will be isolated to only a subframe (usually about 5ms) of speech. For instance, after overflow in a subframe due to a large gain being applied, the SLRP computed for the next subframe can be appropriately set to minimize the likelihood of continued overflows.

-38-

This near-zero buffering delay method is especially applicable to the FR codec since the decoding of the SLRP for this codec does not involve decoding any other parameters. In the case of the EFR codec, the subframe excitation vector is also needed to decode the SLRP and the more complex differential requantization techniques have to be used for requantizing the SLRP. Even in this case, significant reduction in the delay is attained by performing the speech level update based on the current subframe after the SLRP is replaced for the current subframe.

Performing coded domain ALC in conjunction with the proposed TFO standard in GSM networks was discussed above. Under TFO, the transmissions between the handsets and base stations are coded, requiring less than 2 bits per speech sample. However, 8 bits per speech sample are still available for transmission between the base stations. At the base station, the speech is decoded and then A-law companded so that 8 bits per sample are necessary. However, the original coded speech bits are used to replace the 2 least significant bits (LSBs) in each 8-bit A-law companded sample. Once TFO is established between the handsets, the base stations only send the 2 LSBs in each 8-bit sample to their respective handsets and discard the 6 MSBs. Hence vocoder tandeming is avoided.

According to the TFO standard, the received bitstream can be divided into 8-bit samples. The 2 least significant bits of each sample will contain the coded speech bits while the upper 6 bits will contain the bits corresponding to the appropriate PCM samples. Hence a noisy version of the linear speech samples is available to the ALC device in this case. It is possible to use this noisy linear domain speech signal to perform the necessary voice activity, double-talk and speech level measurements as is

-39-

usually done in linear domain ALC algorithms. Thus, in this case, only a minimal amount of decoding of the coded domain speech parameters is necessary. Only parameters that are required for the determination and requantization of the SLRP would have to be decoded. Partial decoding of the speech signal is unnecessary as the noisy linear domain speech samples can be relied upon to measure the speech level as well as perform voice activity and double-talk detection.

Those skilled in communications will recognize that the processes and processing referred to above may be performed by a processor which may include a microprocessor, a microcontroller or a digital signal processor, as well as other logic units capable of logical and arithmetic operations.

Coded Domain ALC In General

Before describing the preferred embodiments, a general discussion of coded domain ALC will be provided. Speech compression, which falls under the category of lossy source coding, is commonly referred to as speech coding. Speech coding is performed to minimize the bandwidth necessary for speech transmission. This is especially important in wireless telephony where bandwidth is scarce. In the relatively bandwidth abundant packet networks, speech coding is still important to minimize network delay and jitter. This is because speech communication, unlike data, is highly intolerant of delay. Hence a smaller packet size eases the transmission through a packet network. The four ETSI GSM standards of concern are listed in Table 3. Each of the standards defines a linear predictive code. Table 3 is a subset of the speech codecs identified in Table 1.

Table 3: GSM Speech Codecs

Codec Name	Coding Method	Bit Rate (kbits/sec)
Half Rate (HR)	VSELP	5.6
Full Rate (FR)	RPE-LTP	13
Enhanced Full Rate (EFR)	ACELP	12.2
Adaptive Multi-Rate (AMR)	MR-ACELP	5.4-12.2

In speech coding, a set of consecutive digital speech samples is referred to as a speech frame. The GSM coders operate on a frame size of 20ms (160 samples at 8kHz sampling rate). Given a speech frame, a speech encoder determines a small set of parameters for a speech synthesis model. With these speech parameters and the speech synthesis model, a speech frame can be reconstructed that appears and sounds very similar to the original speech frame. The reconstruction is performed by the speech decoder. In the GSM speech coders listed above, the encoding process is much more computationally intensive than the decoding process.

The speech parameters determined by the speech encoder depend on the speech synthesis model used. The GSM coders in Table 3 utilize linear predictive coding (LPC) models. A block diagram of a simplified view of the LPC speech synthesis model is shown in Figure 3. The Figure 3 model can be used to generate speech-like signals by specifying the model parameters appropriately. In this example speech synthesis model, the parameters include the time-varying filter coefficients, pitch periods, codebook vectors and the gain factors. The synthetic speech is generated as follows. An appropriate codebook vector, $c(n)$, is first scaled by the codebook gain factor G. Here n denotes sample time. The scaled codebook vector is then filtered by a pitch synthesis filter whose parameters include the pitch gain, g_p, and the pitch period, T. The result is sometimes referred to as the total excitation

vector, $u(n)$. As implied by its name, the pitch synthesis filter provides the harmonic quality of voiced speech. The total excitation vector is then filtered by the LPC synthesis filter which specifies the broad spectral shape of the speech frame.

For each speech frame, the parameters are usually updated more than once. For instance, in the GSM FR and EFR coders, the codebook vector, codebook gain and the pitch synthesis filter parameters are determined every subframe (5ms). The LPC synthesis filter parameters are determined twice per frame (every 10ms) in EFR and once per frame in FR.

A typical speech encoder executes the following sequence of steps:

1. Obtain a frame of speech samples.

2. Multiply the frame of samples by a window (e.g. Hamming window) and determine the autocorrelation function up to lag M .

3. Determine the LPC coefficients from the autocorrelation function.

4. Transform LPC coefficients to a different form (e.g. log-area ratios or line spectral frequencies)

5. Quantize the transformed LPC coefficients using vector quantization techniques.

6. The following sequence of operations is typically performed for each subframe:

7. Determine the pitch period.

8. Determine the corresponding pitch gain.

9. Quantize the pitch period and pitch gain.

10. Inverse filter the original speech signal through the quantized LP synthesis filter to obtain the LP residual signal.

11. Inverse filter the LP residual signal through the pitch synthesis filter to obtain the pitch residual.

12. Determine the best codebook vector.

13. Determine the best codebook gain.

14. Quantize the codebook gain and codebook vector.

15. Update the filter memories appropriately.

16. Transmit the coded parameters.

A typical speech decoder executes the following sequence of steps:

1. Dequantize all the received coded parameters (LPC coefficients, pitch period, pitch gain, codebook vector, codebook gain).

2. Scale the codebook vector by the codebook gain and filter it using the pitch synthesis filter to obtain the LP excitation signal.

3. Filter the LP excitation signal using the LP synthesis filter to obtain a preliminary speech signal.

4. Construct a post-filter (usually based on the LPC coefficients).

5. Filter the preliminary speech signal to reduce quantization noise to obtain the final synthesized speech.

Although many non-linearities and heuristics are involved in the synthesis, the following approximate transfer function may be attributed to the synthesis process which is sufficiently accurate for the purposes of ALC:

$$H(z) = \frac{G}{\left(1 - g_p z^{-T}\right)\left(1 - \sum_{k=1}^{p} a_k z^{-k}\right)} \tag{11}$$

We can consider the codebook vector, $c(n)$, as being filtered by $H(z)$ to result in the synthesized speech. The key point to note is that G specifies the DC gain of the transfer function. This, in turn, implies that G can be modified to adjust the overall speech level in an approximately linear manner. Hence, G is termed the Speech Level Related Parameter (SLRP).

As previously explained in connection with Table 2, GSM coders use speech level related parameters (SLRPs). These SLRPs correspond to G in the general speech synthesis model of Figure 3. To perform coded domain ALC (CD-ALC) in conjunction with a given codec, only the corresponding SLRP needs to be modified in the bit-stream received at the network ALC device. This has the advantage that the re-

-44-

encoding process is greatly simplified. Furthermore, this approach results in the least possible amount of perceptually significant quantization noise being introduced in the signal. For each codec, a different coded domain SLRP modification algorithm must be devised. Here, preferred algorithms for the FR and EFR coders are described.

As previously explained in connection with Figure 6-10, the quantization of a single speech parameter is termed scalar quantization. When a set of parameters are quantized together, the process is called vector quantization. Vector quantization is usually applied to a set of parameters that are related to each other in some way such as the LPC coefficients. Scalar quantization is generally applied to a parameter that is relatively independent of the other parameters, such as the codebook gain. For the purposes of implementing CD-ALC, the discussion is limited to scalar quantization only.

Both the FR and EFR coders utilize scalar quantization for their respective codebook gains (which we are also referring to as the SLRPs). The FR coder performs instantaneous scalar quantization on the SLRP (x_{max}). That is, only the current value of the SLRP is used in the quantization process, which is a relatively simple table look-up method. The EFR coder performs an adaptive differential scalar quantization of the SLRP (γ_{gc}). In this method, the current quantized value depends on past quantized values.

A preferred embodiment of the invention utilizing a modular approach to CD-ALC is shown in Figure 24. A communications system 10 transmits near end digital signals from a near end handset 12 over a network 14 using a compression code, such

as any of the codes used by the Codecs identified in Table 2. The compression code is generated by an encoder 16 from linear audio signals generated by the near end handset 12. The compression code comprises parameters, such as the parameters labeled SLRP in Table 2. The parameters represent an audio signal comprising a plurality of audio characteristics, including audio level. As previously explained, the audio level is related to the parameters labeled SLRP in Table 2. The compression code is decodable by various decoding steps, including one or more steps for decoding the parameters related to audio level. As will be explained, system 10 adjusts the audio level with minimal delay and minimal, if any, decoding of the compression code parameter relating to audio level.

Near end digital signals using the compression code are received on a near end terminal 20 and send in port Sin, and an adjusted compression code is transmitted by a near end terminal 22 and send out port Sout over a network 24 to a far end handset 26 which includes a decoder 28 of the compression code. A linear far end audio signal is encoded by a far end encoder 30 to generate far end digital signals using the same compression code as encoder 16, and is transmitted over a network 32 to a far end terminal 34 and receive in port Rin. Network 34 also transmits the far end signals to a terminal 36 and a receive out port Rout. A decoder 18 of near end handset 12 decodes the far end digital signals. As shown in Figure 24, echo signals from the far end signals may find their way to encoder 16 of the near end handset 12.

A processor 40 performs various operations on the near end and far end compression code. Processor 40 may be a microprocessor, microcontroller, digital

-46-

signal processor, or other type of logic unit capable of arithmetic and logical operations.

For each type of codec, a different coded domain SLRP modification algorithm is executed by processor 40. A linear domain level control algorithm 42 executed by processor 40 is in operation at all times – under native mode and linear mode, during TFO as well as non-TFO. A partial decoder 48 decodes enough of the compression code to form linear code from which the audio level of the audio signal represented by the compression code can be determined. Decoder 48 also reads a compression code parameter related to audio level, such as one of the parameters identified in Table 2. The read parameter is dequantized to form a parameter value. The linear domain level control algorithm determines the gain factor for level adjustment and writes it to a predetermined memory location within processor 40. This gain factor is read by the appropriate codec-dependent coded domain SLRP modification algorithm 44 also executed by processor 40. Algorithm 44 modifies the read SLRP parameter (i.e., the gain factor) to form an adjusted SLRP parameter value (i.e., adusted gain factor). The adjusted parameter value is quantized to form an adjusted SLRP parameter which is written into the bit-stream received at terminal 20. In other words, the adjusted SLRP parameter is substitued for the original read SLRP paramter. The partial decoders 46 and 48 shown within the Network ALC Device are algorithms executed by processor 40 and are codec-dependent. In the case of GSM EFR, the decoder post-filtering operations except for upscaling are unnecessary. In the case of GSM FR, the complete decoder is implemented.

A modular approach has the advantage that any existing or new linear domain level control algorithm can be incorporated with little or no modification with the coded domain SLRP modification algorithms. A coder-specific level control method might provide more accurate level adjustments. However, it may require a significant re-design of the existing linear domain level control algorithms to ensure smooth transitions when switching from native to linear mode (and vice versa). Note that there is a small risk that some undesirable artifacts may be occasionally introduced when switching between coded and linear modes when using the modular approach.

The preferred embodiment includes a minimal delay technique. Large buffering, processing and transmission delays are already present in cellular networks without any network voice quality enhancement processing. Further network processing of the coded speech for speech enhancement purposes will add additional delay. If linear domain processing is performed on coded speech during TFO, more than a frame of delay (20ms) will be added due to buffering and processing requirements for decoding and re-encoding. However, CD-ALC can be performed with a buffering delay that is much less than one frame for FR and EFR coders.

The delay reduction under CD-ALC is achieved for FR and EFR by performing level control a subframe at a time rather than frame-by-frame. As soon as a subframe is decoded by decoder 48 and the necessary level measurements are updated, the linear domain ALC algorithm can send the gain factor to the coded domain SLRP modification algorithm 44. Due to the manner in which the parameters are arranged in the received bit-stream, the first subframe requires more than 5ms of delay before decoding can begin.

Table 5 and Table 6 provide the earliest possible points at which decoding of samples can be performed as the bit-stream is received for the FR and EFR coders, respectively, and correspond to the illustration in Figure 23. Note that there are 260 bits/frame for the FR and 244 bits/frame for the EFR. The table assumes that the incoming bits are spread out evenly over 20ms, for the sake of simplicity. With this approximation, the first subframe requires 7.11ms for the FR and 7.46ms for the EFR. All other subframes require less delay.

Table 5: Earliest possible decoding of samples in the GSM FR coder

Bits Received	Delay from first bit (ms)	Decodable Samples
1-92	7.11	1-40
93-148	11.4	41-80
149-204	15.7	81-120
205-260	20.0	121-160

Table 6: Earliest possible decoding of samples in the GSM EFR coder

Bits Received	Delay from first bit (ms)	Decodable Samples
1-91	7.46	1-40
92-141	11.6	41-80
142-194	15.9	81-120
195-244	20.0	121-160

CD-ALC For GSM FR

For the purposes of CD-ALC for GSM FR, we are concerned only with the modification of the SLRP parameter called the block maximum, x_{max} (seeTable 2). This parameter corresponds to G in the speech synthesis transfer function given by equation (11). This section explains the decoding of this parameter from the 260 bits received each frame. (Refer to the "RPE Encoding Section" of Reference [1] (sections 3.1.18-3.1.22) for a functional description of the determination of x_{max}. The

corresponding pseudo-code for determining x_{max} is found in sections 4.2.13-4.2.17 of Reference[1].)

In the 260 bits received in each frame, the specific bits from which x_{max} can be determined are described in Table7. The six bits indicated for each subframe are used as the index into a 64-word table specified by Table 3.5, "Quantization of the block maximum, x_{max} ", in [1]. In Table 7, the index is denoted by $x_{max c}$ and the corresponding value is denoted by x'_{max}.

Table7: FR Encoder Block Maximum bit positions within speech frame of 260 bits/20ms

Subframe	Variable name	Bit no. (LSB-MSB)
1	x_{max1}	48-53
2	x_{max2}	104-109
3	x_{max3}	160-165
4	x_{max4}	216-221

For encoding (i.e., quantization of) the SLRP parameter after modification, Table 3.5, "Quantization of the block maximum, x_{max} ", in Reference [1] is used. The table specifies a six bit index for each range of values. The six bit index is re-inserted in the appropriate positions for each subframe.

The quantized SLRP values are shown in Figure 6. The range of the quantized values is 31 to 32767. This represents a dynamic range of about 60dB $(20\log_{10}(32767/31))$.

The processing of each subframe of the SLRP is as follows:

-50-

(1) Both the near-end and far-end compression coded speech subframes are fully decoded by decoders 46 and 48. That is, the digital signals transmitted to terminals 20 and 34 are both fully decoded by decoders 46 and 48 to generate near end decoded signals and far end decoded signals indicative of audio level. In addition, the x'_{max} value is read from the coded near end signal by partial decoder 48. (Alignment of subframe boundaries between the two ends is not important.) The near end decoded signals and far end decoded signals are processed by the Linear Domain ALC (LD-ALC) algorithm 42 to determine the proper audio level. Depending on the implementation, only the double-talk information based on the far-end signal received at terminal 34 may be actually passed into the LD-ALC algorithm 42.

(2) The current subframe of the near-end signal (Sin port) is scaled by LD-ALC 42.

(3) The LD-ALC gain or level, denoted by g_{ALC}, used for processing the last sample of the current subframe is passed into CD-ALC 44. This may be achieved by writing to a predetermined memory location to be read by CD-ALC.

(4) CD-ALC 44 extracts the 6-bit table index for the current subframe according to Table 7 above. The quantized x'_{max} value is then determined using Table 3.5, "Quantization of the block maximum, x_{max}", in Reference [1]. Alternatively, since the decoder has already looked up this value, the decoder code may be modified to pass this value to CD-ALC 44.

(5) A new block maximum (adjusted level value) is computed as $x_{max,new} = g_{ALC} \times x'_{max}$.

(6) $x_{max,new}$ is quantized using Table 3.5, "Quantization of the block maximum, x_{max}", in Reference [1]. The resulting 6-bit table index which represents an adjusted level

parameter is inserted (e.g., written or substituted) appropriately back into the coded near end bit-stream according to Table7.

(7)   Any CRC or error control coding bits are updated appropriately.

CD-ALC For GSM EFR

A preferred form of CD-ALC for GSM EFR will be explained. The quantization of the SLRP in the GSM EFR coder is not as straightforward as the FR. Hence an overview of the encoding and decoding of the SLRP is first provided.

For the purposes of CD-ALC, we are concerned only with the modification of the parameter called the codebook gain, g_c (Table 2). This parameter corresponds to G in the speech synthesis transfer function given by equation (11). However, this parameter is not directly available in the received bit-stream. A rather indirect form of adaptive differential quantization using a static linear predictor is utilized for quantizing g_c every subframe. The 'gain' parameter that is transmitted is actually a correction factor between g_c and the predicted gain, g_c'. This correction factor, γ_{gc}, is defined as

$$\gamma_{gc} = g_c / g_c' \qquad (12)$$

γ_{gc} is considered the actual compression code SLRP because it is the only parameter related to the overall speech level that is accessible directly in the coded domain.

-52-

At the encoder (e.g., encoder 16), once the best g_c for the current subframe is determined, it is divided by the predicted gain g_c' to obtain γ_{gc}. The predicted gain for subframe n is given by

$$g_c'(n) = 10^{0.05[\tilde{E}(n) - E_I(n) + \bar{E}]} \tag{13}$$

A 32-level non-uniform quantization is performed on γ_{gc} to obtain $\hat{\gamma}_{gc}$. The encoder transmits the look-up table index corresponding to $\hat{\gamma}_{gc}$. In (13), $\bar{E}$ is a constant, $E_I(n)$ depends only on the subframe's fixed codebook vector, and $\tilde{E}(n)$ depends only on the previously quantized correction factors. The decoder, thus, calculates the predicted gain g_c' in the same manner as the encoder using (13) once the current subframe's fixed codebook vector is decoded. On decoding the correction factor $\hat{\gamma}_{gc}$, the quantized gain factor is computed using (12) as

$$\hat{g}_c(n) = \hat{\gamma}_{gc}(n) \times g_c'(n) \tag{14}$$

The adaptive differential quantization of the SLRP, γ_{gc}, is performed in the logarithmic domain. The process is illustrated in Figure 25 in which $R(n)$ denotes the prediction error given by $R(n) = E(n) - \tilde{E}(n) = 20 \log \gamma_{gc}(n)$. $R(n)$ is quantized by the block denoted by Q in the figure to $\hat{R}(n)$; the quantization is performed using a 32-word quantization table for $\hat{\gamma}_{gc}$ given in the array "qua_gain_code" specified in the bit-true C code file "gains_tb.h" that comes with the EFR standard described in Reference[2]. This array is reproduced in Table 9 below.

-53-

The same static linear predictor, $P(z)$, with fixed coefficients is used at both the encoder and decoder; it is given by $P(z) = 0.68z^{-1} + 0.58z^{-2} + 0.34z^{-3} + 0.19z^{-4}$.

The quantization of the SLRP at the encoder is performed indirectly by using the mean-removed codebook vector energy each subframe. $E(n)$ denotes the mean-removed codebook vector energy (in dB) at subframe n and is given by

$$
\begin{aligned}
E(n) &= 10\log\left[\frac{1}{40}g_c^2\sum_{i=0}^{39}c^2(i)\right] - \bar{E} \\
&= 20\log g_c + 10\log\left[\frac{1}{40}\sum_{i=0}^{39}c^2(i)\right] - \bar{E} \qquad (15) \\
&= 20\log g_c + E_t(n) - \bar{E}
\end{aligned}
$$

where the mean codebook vector energy is given by

$$
E_t(n) = 10\log\left[\frac{1}{40}\sum_{i=0}^{39}c^2(i)\right] \qquad (16)
$$

The codebook vector $\{c(i)\}$ is required in order to decode the SLRP. Note that the decoding of the codebook vector is independent of the decoding of the SLRP. We see that $E(n)$ is a function of the gain factor, g_c. The quantization of γ_{gc} to $\hat{\gamma}_{gc}$ indirectly results in the quantization of g_c to $\hat{g}_c$. This quantized gain factor is used to scale the codebook vector, hence setting the overall level of the audio signal synthesized at the decoder (e.g., decoder 28). $\tilde{E}(n)$ is the predicted energy given by

$$
\tilde{E}(n) = 0.68\hat{R}(n-1) + 0.58\hat{R}(n-2) + 0.34\hat{R}(n-3) + 0.19\hat{R}(n-4) \qquad (17)
$$

where $\{\hat{R}(n-i)\}$ are previously quantized values.

A summary of the process of decoding the codebook gain factor, g_c, follows. First, the decoder decodes the excitation vector and computes $E_t(n)$ using (16).

Second, $\tilde{E}(n)$ is computed using previously decoded gain correction factors using (17). Then the predicted gain g_c' is computed using (13). Next, the received index of the correction factor for the current subframe is used to obtain $\hat{\gamma}_{gc}$ from the look-up table. Finally, the quantized gain factor is obtained via (14).

In the 244 bits received each frame, the specific bits from which $\hat{\gamma}_{gc}$ can be determined are specified in Table 8. The five bits indicated for each subframe are used as the index into a 32-word array "qua_gain_code" specified in the bit-true C code file "gains_tb.h" that comes with the EFR standard described Reference[2]. This information is also provided in Table 9.

Table 8: EFR Encoder Codebook Gain Parameter bit positions within speech frame of 244 bits/20ms

Subframe	Bit no. (LSB-MSB)
1	87-91
2	137-141
3	190-194
4	240-244

The quantized SLRP values are shown in Figure 10. Differences between adjacent quantization levels are shown in Figure 22. The range of the quantized values is 159 to 27485. This represents a dynamic range of about 45dB $(20\log_{10}(27485/159))$. The table of quantized SLRP values and the logarithms are also provided in Table 9. This table is necessary for re-encoding the SLRP.

Table 9: Table of SLRP quantization values for GSM EFR

Index	$\hat{\gamma}_{gc}$	$\hat{R}(n) = 20\log\hat{\gamma}_{gc}(n)$
0	159	44.027942
1	206	46.277344

2	268	48.562696
3	349	50.856509
4	419	52.444280
5	482	53.660941
6	554	54.870195
7	637	56.082789
8	733	57.302079
9	842	58.506242
10	969	59.726476
11	1114	60.937704
12	1281	62.150983
13	1473	63.364055
14	1694	64.578268
15	1948	65.791779
16	2241	67.008837
17	2577	68.222288
18	2963	69.434633
19	3408	70.649992
20	3919	71.863505
21	4507	73.077751
22	5183	74.291624
23	5960	75.504925
24	6855	76.720149
25	7883	77.933831
26	9065	79.147356
27	10425	80.361521
28	12510	81.945146
29	16263	84.224013
30	21142	86.502921
31	27485	88.781915

CD-ALC Processing of the SLRP

The CD-ALC processing of the SLRP of each subframe is as follows:

(1) Both the near-end and far-end compression coded speech subframes are fully

decoded by decoders 46 and 48. That is, the digital signals transmitted to

terminals 20 and 34 are both fully decoded by decoders 46 and 48 to generate near

end decoded signals and far end decoded signals. In addition, the γ_{gc} parameter is

read from the coded near end signal by partial decoder 48. (Alignment of

subframe boundaries between the two ends is not important.) The near end
decoded and far end decoded signals are processed by the Linear Domain ALC
(LD-ALC) algorithm to determine the proper audio level. Depending on the
implementation, only the double-talk information based on the far-end signal may
5   be actually passed into the LD-ALC algorithm 42.

(2) The current subframe of the near-end signal (Sin port) is scaled by LD-ALC 42.

(3) The LD-ALC gain or level, denoted by g_{ALC}, used for processing the last sample
of the current subframe is passed into CD-ALC 44. This may be achieved by
writing to a predetermined memory location to be read by CD-ALC.

10   (4) CD-ALC 44 extracts the 5-bit table index for the current subframe according to
Table 8 above. Alternatively, since the decoder has already determined this index,
the decoder code may be modified to pass this value to CD-ALC 44.

(5) The 5-bit table index, Table 9, is used to determine $\hat{R}(n) = 20\log_{10}\left(\hat{\gamma}_{gc}\right)$ which is a
dequantized parameter value.

15   (6) A table look-up is performed to determine $20\log_{10}\left(g_{ALC}\right)$. This is possible since
the possible values that g_{ALC} can take on are predetermined, and hence can be
precomputed.

(7) $R_{new}(n)$ denotes the new or adjusted SLRP value. Four variables,
$\{PastDeltaR[0], PastDeltaR[1], PastDeltaR[2], PastDeltaR[3]\}$, which must be
20   kept in memory from one subframe to the next are also required. These variables
are initialized to zero at the beginning of a call.

(8) The predicted dB gain, $Gain_{predicted}(n)$, is computed as

$$Gain_{predicted}(n) =$$
$$0.68 * PastDeltaR[0]$$
$$+0.58 * PastDeltaR[1] \qquad\qquad (18)$$
$$+0.34 * PastDeltaR[2]$$
$$+0.19 * PastDeltaR[3]$$

(9) The actual unquantized gain or level, $Gain_{actual}(n)$ then is computed as the

difference between the desired and predicted gains as follows:

$$Gain_{actual}(n) = 20\log_{10} g_{ALC}(n) - Gain_{predicted}(n) \qquad\qquad (19)$$

(10)   The state of the predictor is updated for use with the next subframe:

$$PastDeltaR[3] = PastDeltaR[2]$$
$$PastDeltaR[2] = PastDeltaR[1]$$
$$PastDeltaR[1] = PastDeltaR[0] \qquad\qquad (20)$$
$$PastDeltaR[0] = Gain_{actual}(n)$$

(11) $R_{new}(n) = \hat{R}(n) + Gain_{actual}(n)$ is computed.

(12) $R_{new}(n)$ is quantized to obtain an adjusted parameter $\hat{R}_{new}(n)$ using Table 9.

This is done by comparing $R_{new}(n)$ to the 32 possible values of $\hat{R}(n)$ in Table 9.

$\hat{R}_{new}(n)$ is assigned the value that is closest in terms of the absolute difference

between $R_{new}(n)$ and a table value. The 5-bit table index corresponding to

$\hat{R}_{new}(n)$ is inserted (e.g., written or substituted) appropriately back into the coded

near end bit-stream according to Table 8.

(13)   Any CRC or error control coding bits are updated appropriately.

Referring to Figure 26, the reasoning behind the re-encoding scheme is

described.

From (15), we know that $E(n) = 20\log g_c + E_I(n) - \overline{E}$ at the encoder. We redraw Figure 25 to explicitly show this in Figure 26.

Suppose ALC is performed prior to encoding. Then $20\log g_c$ is replaced by $20\log(g_c \times g_{ALC}) = 20\log g_c + 20\log g_{ALC}$ in the SLRP encoding process. Since our goal is to perform ALC in the network which has no access to the original encoder, the encoding process is mimicked in the network as shown in Figure 27. Except for the quantizer, the process at the encoder is a linear system with transfer function $1/[1+P(z)]$. The process at the CD-ALC Device also has this linear transfer function. The outputs of these two processes are added and the resulting sum is denoted by $R_{new}(n)$. $R_{new}(n)$ is approximately equal to the ideal ALC-processed value of $20\log(g_c \times g_{ALC})$. $R_{new}(n)$ is quantized to $\hat{R}_{new}(n)$ so that the look-up table index can be re-inserted into the bit-stream. This is the method specified in the CD-ALC Processing of the SLRP section.

In the ALC application, the gain factor changes are generally small and infrequent relative to the subframe rate. This implies that $20\log g_{ALC}$ is kept constant for a large number of subframes. Since the order of $P(z)$ is small, the output of the process $1/[1+P(z)]$ reaches steady state in a relatively small number of subframes. Thus, it seems reasonable to approximate the process $1/[1+P(z)]$ by $1/[1+P(1)] = 1/2.79$. With this approximation, we can compute $R_{new}(n) = \hat{R}(n) + \dfrac{20\log_{10}(g_{ALC})}{2.79}$, which is simpler than the procedure in CD-

-59-

ALC Processing of the SLRP section. However, larger transients may be observed with this method for some applications.

Modifications to LD-ALC Algorithms

The following modifications of the LD-ALC algorithm (e.g. TLC) are preferred for smooth transitioning between linear and native mode processing (e.g. in the case of handovers):

(1)   The gain factor adjustment steps should be limited to ±3 dB for operation in conjunction with GSM FR codecs, which is the same as the usual LD-ALC step size. (In some version of LD-ALC, 6dB steps were possible; this should be avoided.) Hence the possible dB gain values should be restricted to {-3, -6, 0, 3, 6, 9, 12, 15}.

(2)   The gain factor adjustment steps should be limited to ±3.39 dB steps for operation in conjunction with GSM EFR codecs. (In some version of LD-ALC, 6dB steps were possible; this should be avoided.) This step size is optimized specifically for EFR to minimize the transient effects and maximize accuracy. Hence the possible dB gain values should be restricted to {-6.77, -3.39, 0, 3.39, 6.77, 10.16, 13.55, 16.93}.

The following are recommended to further enhance performance:

(1)   Any gain changes should be restricted to occur only at the beginning of a subframe boundary. This ensures that the sample at which a gain change occurs is identical in both the linear (upper 6 PCM bits) and coded signals.

     (2)   A subframe (40 samples) of speech should be processed at a time for efficiency.

An Example Of CD-ALC Results

Since the CD-ALC algorithm utilizes an LD-ALC algorithm to determine the gain adjustments, the CD-ALC algorithm performance is, in a sense, upper bounded by the LD-ALC performance. Thus, even if the LD-ALC algorithm complies with G.169, Reference [3], the CD-ALC algorithm should also be tested for G.169 compliance.

In this section, typical level adjustment results are illustrated. The exemplary speech signal used is illustrated in Figure 28.

Figure 29 shows the results for a case when CD-ALC is used in conjunction with FR. The upper plot shows power profiles of the original (dashed line) and processed (solid line) signals. A 40ms time constant was used in the recursive mean-square averaging of the signals to obtain the power profiles. The lower plot shows the LD-ALC gain (blue, dashed line) at the end of each subframe; also shown is the ratio of the processed power to the original power at the end of each subframe. In the regions where the speech signal is strong, the amplification of the signal corresponds quite closely to the desired gain.

Those skilled in the art of communications will recognize that the preferred embodiments can be modified and altered without departing from the true spirit and scope of the invention as defined in the appended claims.

What is claimed is:

1.   In a communications system for transmitting digital signals using a compression code comprising a predetermined plurality of parameters including a first parameter, said parameters representing an audio signal comprising a plurality of audio characteristics including a first characteristic, said first parameter being related to said first characteristic, said compression code being decodable by a plurality of decoding steps including a first decoding step for decoding said parameters related to said first characteristic, apparatus for adjusting the first characteristic comprising:

a processor responsive to said digital signals to read at least said first parameter and to generate at least a first parameter value derived from said first parameter, responsive to said digital signals and said first parameter value to generate an adjusted first parameter value representing an adjustment of said first characteristic, and responsive to said adjusted first parameter value to derive an adjusted first parameter and to replace said first parameter with said adjusted first parameter.

2.   Apparatus, as claimed in claim 1, wherein said first characteristic comprises a level of said audio signal.

3.   Apparatus, as claimed in claim 1, wherein said plurality of decoding steps further comprise at least one decoding step avoiding substantial altering of the first characteristic and wherein said processor avoids performing said at least one decoding step.

4.   Apparatus, as claimed in claim 3, wherein said at least one decoding step comprises post-filtering.

-62-

5.   Apparatus, as claimed in claim 1, wherein said compression code comprises a linear predictive code.

6.   Apparatus, as claimed in claim 1, wherein said compression code comprises regular pulse excitation – long term prediction code.

7.   Apparatus, as claimed in claim 6, wherein said digital signals are transmitted in frames comprising subframes and wherein said first parameter comprises a maximum absolute value of the elements in a codebook vector for one of said subframes.

8.   Apparatus, as claimed in claim 1, wherein said compression code comprises algebraic code-excited linear prediction code.

9.   Apparatus, as claimed in claim 8, wherein said digital signals are transmitted in frames comprising subframes, wherein said first parameter comprises a gain correction factor for one of said subframes.

10.   Apparatus, as claimed in claim 1, wherein said digital signals comprise a near end digital signal using a near end compression code comprising a predetermined plurality of near end parameters including a first near end parameter, said near end parameters representing a near end audio signal comprising a plurality of near end audio characteristics including a near end first characteristic, said near end first parameter being related to said near end first characteristic, said near end compression code being decodable by a plurality of decoding steps including a first decoding step for decoding said near end parameters related to said near end first characteristic, said digital signals further comprising a far end digital signal using a far end compression code comprising a predetermined plurality of far end parameters, said far end parameters representing a far end audio signal comprising a plurality of far end audio characteristics including a far

-63-

end first characteristic, said far end compression code being decodable by a plurality of

decoding steps including a first decoding step for decoding said far end parameters

related to said far end first characteristic,

wherein said processor receives said near end digital signal and said far end

5   digital signal,

wherein said processor is responsive to said near end digital signal to read

at least said near end first parameter and to generate a near end first parameter value

derived from said near end first parameter,

wherein said processor is responsive to said near end digital signal to

10   perform at least said first decoding step to generate near end decoded signals related to

said near end first characteristic of said near end audio signal,

wherein said processor is responsive to said far end digital signal to

perform at least said first decoding step to generate far end decoded signals related to

said far end first characteristic of said far end audio signal,

15   wherein said processor is responsive to said near end decoded signals, said

far end decoded signals and said near end first parameter value to generate an adjusted

near end first parameter value representing an adjustment of said near end first

characteristic,

wherein said processor derives an adjusted near end first parameter from

20   said adjusted near end first parameter value, and

wherein said processor replaces said near end first parameter with said

adjusted near end first parameter.

11.   Apparatus, as claimed in claim 1, wherein said processor tests said adjusted first parameter value for an overflow and underflow condition before deriving said adjusted first parameter.

12.   Apparatus, as claimed in claim 11, wherein said first parameter is a quantized first parameter and wherein said processor derives said adjusted first parameter by quantizing said adjusted first parameter value.

13.   Apparatus, as claimed in claim 12, wherein said processor uses differential scalar quantization during said quantizing.

14.   Apparatus, as claimed in claim 13, wherein said processor uses differential scalar quantization with a quantizer outside feedback loop during said quantizing.

15.   Apparatus, as claimed in claim 1, wherein said first parameter comprises a series of first parameters received over time, wherein said processor is responsive to said digital signals to read said series of first parameters and to generate a series of first parameter values over time, and wherein said processor is responsive to said decoded signals and to at least a plurality of said series of first parameter values to generate said adjusted first parameter value.

16.   Apparatus, as claimed in claim 15, wherein said first parameter is a quantized first parameter and wherein said processor derives said adjusted first parameter by quantizing said adjusted first parameter value.

17.   Apparatus, as claimed in claim 16, wherein said processor uses differential scalar quantization during said quantizing.

18.   Apparatus, as claimed in claim 1, wherein said first parameter is a quantized first parameter and wherein said processor derives said adjusted first parameter by quantizing said adjusted first parameter value.

19.   Apparatus, as claimed in claim 18, wherein said processor uses differential

scalar quantization during said quantizing.

20.   Apparatus, as claimed in claim 18, wherein said processor performs said

quantizing using instantaneous scalar quantization techniques.

21.   Apparatus, as claimed in claim 1, wherein said compression code is

arranged in frames of said digital signals and wherein said frames comprise a plurality of

subframes each comprising said first parameter, wherein said processor is responsive to

said digital signals to read at least said first parameter from each of said plurality of

subframes, and wherein said processor replaces said first parameter with said adjusted

first parameter in each of said plurality of subframes.

22.   Apparatus, as claimed in claim 21, wherein said processor replaces said

first parameter with said adjusted first parameter for a first subframe before processing a

subframe following the first subframe to achieve lower delay.

23.   Apparatus, as claimed in claim 1, wherein said compression code is

arranged in frames of said digital signals and wherein said frames comprise a plurality of

subframes each comprising said first parameter, wherein said processor performs at least

said first decoding step during a first of said subframes to generate said decoded signals,

reads said first parameter from a second of said subframes occurring subsequent to said

first subframe to generate said first parameter value, generates said adjusted first

parameter value in response to said decoded signals and said first parameter value, and

replaces said first parameter of said second subframe with said adjusted first parameter.

24.   Apparatus, as claimed in claim 1, wherein said processor performs at least

said first decoding step to generate decoded signals related to said first characteristic of

-66-

said audio signal and wherein said processor is responsive to said decoded signals and said first parameter value to generate said adjusted first parameter value.

25.   In a communications system for transmitting digital signals comprising code samples, said code samples comprising first bits using a compression code and second bits using a linear code, said code samples representing an audio signal, said audio signal having a plurality of audio characteristics including a first characteristic, apparatus for adjusting the first characteristic without decoding said compression code comprising:

a processor responsive to said second bits to adjust said first bits and said second bits, whereby said first characteristic is adjusted.

26.   Apparatus, as claimed in claim 25, wherein said linear code comprises pulse code modulation (PCM) code.

27.   Apparatus, as claimed in claim 25, wherein said first characteristic comprises audio level.

28.   Apparatus, as claimed in claim 25, wherein said compression code samples conform to the tandem-free operation of the global system for mobile communications standard.

29.   Apparatus, as claimed in claim 25, wherein said first bits comprise the two least significant bits of said samples and wherein said second bits comprise the 6 most significant bits of said samples.

30.   Apparatus, as claimed in claim 29, wherein said 6 most significant bits comprise PCM code.

31.   In a communications system for transmitting digital signals using a compression code comprising a predetermined plurality of parameters including a first

-67-

parameter, said parameters representing an audio signal comprising a plurality of audio

characteristics including a first characteristic, said first parameter being related to said

first characteristic, said compression code being decodable by a plurality of decoding

steps including a first decoding step for decoding said parameters related to said first

5   characteristic, a method of adjusting the first characteristic comprising:

reading at least said first parameter in response to said digital signals;

generating at least a first parameter value derived from said first parameter;

performing at least said first decoding step to generate decoded signals

related to said first characteristic of said audio signal in response to said digital signals;

10   generating an adjusted first parameter value representing an adjustment of

said first characteristic in response to said digital signals and said first parameter value;

deriving an adjusted first parameter in response to said adjusted first

parameter value; and

replacing said first parameter with said adjusted first parameter.

15   32.   A method, as claimed in claim 31, wherein said first characteristic

comprises a level of said audio signal.

33.   A method, as claimed in claim 31, wherein said plurality of decoding steps

further comprise at least one decoding step avoiding substantial altering of the first

characteristic and wherein said method avoids performing said at least one decoding

20   step.

34.   A method, as claimed in claim 33, wherein said at least one decoding step

comprises post-filtering.

35.   A method, as claimed in claim 31, wherein said compression code

comprises a linear predictive code.

-68-

36.   A method, as claimed in claim 31, wherein said compression code comprises regular pulse excitation – long term prediction code.

37.   A method, as claimed in claim 36, wherein said digital signals are transmitted in frames comprising subframes and wherein said first parameter comprises a maximum absolute value of the elements in a codebook vector for one of said subframes.

38.   A method, as claimed in claim 31, wherein said compression code comprises code-excited linear prediction code.

39.   A method, as claimed in claim 38, wherein said digital signals are transmitted in frames comprising subframes, wherein said first parameter comprises a gain correction factor.

40.   A method, as claimed in claim 31, wherein said digital signals comprise a near end digital signal using a near end compression code comprising a predetermined plurality of near end parameters including a first near end parameter, said near end parameters representing a near end audio signal comprising a plurality of near end audio characteristics including a near end first characteristic, said near end first parameter being related to said near end first characteristic, said near end compression code being decodable by a plurality of decoding steps including a first decoding step for decoding said near end parameters related to said near end first characteristic, said digital signals further comprising a far end digital signal using a far end compression code comprising a predetermined plurality of far end parameters, said far end parameters representing a far end audio signal comprising a plurality of far end audio characteristics including a far end first characteristic, said far end compression code being decodable by a plurality of

-69-

decoding steps including a first decoding step for decoding said far end parameters related to said far end first characteristic,

wherein said receiving said digital signals comprises receiving said near end digital signal and said far end digital signal,

wherein said reading comprises reading at least said near end first parameter;

wherein said generating a first parameter comprises generating a near end first parameter value derived from said near end first parameter,

wherein said performing at least said first decoding steps comprises generating near end decoded signals related to said near end first characteristic of said near end audio signal in response to said near end digital signal and generating far end decoded signals related to said far end first characteristic of said far end audio signal in response to said far end digital signal,

wherein said generating an adjusted first parameter value comprises generating an adjusted near end first parameter value representing an adjustment of said near end first characteristic in response to said near end decoded signals, said far end decoded signals and said near end first parameter value,

wherein said deriving an adjusted first parameter comprises deriving an adjusted near end first parameter from said adjusted near end first parameter value, and

wherein said replacing comprises replacing said first parameter with said adjusted first parameter.

41.   A method, as claimed in claim 31, and further comprising testing said adjusted first parameter value for an overflow and underflow condition before deriving said adjusted first parameter.

-70-

42. A method, as claimed in claim 41, wherein said first parameter is a quantized first parameter and wherein said deriving an adjusted first parameter comprises quantizing said adjusted first parameter value.

43. A method, as claimed in claim 42, and further comprising using differential scalar quantization during said quantizing.

44. A method, as claimed in claim 43, wherein said using differential scalar quantization comprises using a quantizer outside feedback loop during said quantizing.

45. A method, as claimed in claim 31, wherein said first parameter comprises a series of first parameters received over time, wherein said reading at least said first parameter comprises reading said series of first parameters, wherein said generating a first parameter value comprises generating a series of first parameter values over time, and wherein said generating an adjusted first parameter value comprises generating said adjusted first parameter value in response to said decoded signals and to at least a plurality of said series of first parameter values.

46. A method, as claimed in claim 45, wherein said first parameter is a quantized first parameter and wherein said deriving an adjusted first parameter comprises quantizing said adjusted first parameter value.

47. A method, as claimed in claim 46, and further comprising using differential scalar quantization during said quantizing.

48. A method, as claimed in claim 31, wherein said first parameter is a quantized first parameter and wherein said deriving an adjusted first parameter comprises quantizing said adjusted first parameter value.

49. A method, as claimed in claim 48, and further comprising using differential scalar quantization during said quantizing.

50. A method, as claimed in claim 48, wherein said quantizing comprises using instantaneous scalar quantization techniques.

51. A method, as claimed in claim 31, wherein said compression code is arranged in frames of said digital signals and wherein said frames comprise a plurality of subframes each comprising said first parameter, wherein said reading at least said first parameter comprises reading at least said first parameter from each of said plurality of subframes, and wherein said replacing comprises replacing said first parameter with said adjusted first parameter in each of said plurality of subframes.

52. A method, as claimed in claim 51, wherein said replacing comprises replacing said first parameter with said adjusted first parameter for a first subframe before processing a subframe following the first subframe to achieve lower delay.

53. A method, as claimed in claim 31, wherein said compression code is arranged in frames of said digital signals and wherein said frames comprise a plurality of subframes each comprising said first parameter, wherein said performing at least said first decoding step comprises performing at least said first decoding step during a first of said subframes to generate said decoded signals, wherein said reading at least said first parameter comprises reading at least said first parameter from a second of said subframes occurring subsequent to said first subframe, wherein said generating a first parameter value comprises generating a first parameter value from said first parameter from said second of said subframes, wherein said generating an adjusted first parameter value comprises generating said adjusted first parameter value in response to said decoded signals and said first parameter value, and wherein said replacing comprises replacing said first parameter of said second subframe with said adjusted first parameter.

54.   A method, as claimed in claim 31, wherein said generating an adjusted first parameter comprises performing at least said first decoding step to generate decoded signals related to said first characteristic of said audio signal in response to said compression code and wherein said generating an adjusted first parameter is responsive to said decoded signals and said first parameter value.

55.   In a communications system for transmitting digital signals comprising code samples, said code samples comprising first bits using a compression code and second bits using a linear code, said code samples representing an audio signal, said audio signal having a plurality of audio characteristics including a first characteristic, a method of adjusting the first characteristic without decoding said compression code comprising:

adjusting said first bits and said second bits in response to said second bits, whereby said first characteristic is adjusted.

56.   A method, as claimed in claim 55, wherein said linear code comprises pulse code modulation (PCM) code.

57.   A method, as claimed in claim 55, wherein said first characteristic comprises audio level.

58.   A method, as claimed in claim 55, wherein said compression code samples conform to the tandem-free operation of the global system for mobile communications standard.

59.   A method, as claimed in claim 55, wherein said first bits comprise the two least significant bits of said samples and wherein said second bits comprise the 6 most significant bits of said samples.

60.   A method, as claimed in claim 59, wherein said 6 most significant bits comprise PCM code.

Figure 1: Network enhancement of coded speech in the linear domain

$$\frac{1}{1-g_p z^{-T}}$$

$$\frac{1}{1-\sum_{k=1}^{p} a_k z^{-k}}$$
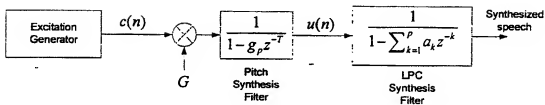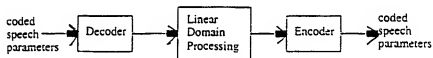
Figure 3

Figure 2
far end signal

ALC Device

level-adjusted near end signal

near end signal

echo

Figure 4
Linear Domain Digital Speech Samples

Speech Encoder

transmission channel or storage medium

Coded Domain

Speech Decoder

Linear domain digital speech samples

Figure 5
far end signal

Speech Encoder 1

Speech Decoder 1

echo

level-adjusted near end signal

Speech Decoder 2

ALC Device

Speech Encoder 2

near end signal

Figure 6

GSM Full Rate Codec Quantization of Block Maxima

Figure 7

(a) Standard Deviation Based

Quantizer

(b) Differential

Quantizer

Dequantizer

Figure 8
  Quantizer

  Dequantizer
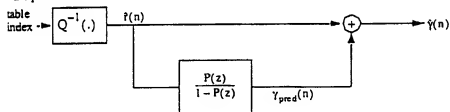
Figure 9

At the Encoder   At the Decoder

Figure 10

GSM Enhanced Full Rate Codec Quantization of Gain Correction Factor

Figure 11

Speech Encoder → Speech Decoder → Linear Domain ALC → Speech Encoder → Speech Decoder

ALC Device

Figure 12

Speech Encoder → Modify SLRP → Speech Decoder

Partial or Complete Decoding → SLRP → ⊗ → SLRP Requantization

gain

Speech Level Measurement → Gain Determination ← Voice Activity and Double-talk information

Figure 13

START → Compute $\hat{g}(n)$

$\hat{\gamma}_{slrp}(n) = Q(\hat{g}(n) \times \hat{\gamma}(n))$

$\tilde{g}(n) = \dfrac{\hat{\gamma}_{slrp}(n)}{\hat{\gamma}(n)}$

Overflow? — No → Underflow? — No → END

Yes ↓   Yes ↓

Reduce $\hat{\gamma}_{slrp}(n)$ to next lower level   Increase $\hat{\gamma}_{slrp}(n)$ to next higher level

$\hat{g}(n)$: desired gain
$\tilde{g}(n)$: realized gain
$\hat{\gamma}(n)$: original (received) SLRP value
$\hat{\gamma}_{slrp}(n)$: new SLRP value
n : subframe or frame index

Figure 14

original SLRP → ⊗ → SLRP Requantization → new SLRP

desired gain

Gain Determination ← realized gain ← compute $\dfrac{new\ SLRP}{original\ SLRP}$

Figure 15

Speech Encoder → Modify SLRP → Speech Decoder

Partial or Complete Decoding

Joint Gain Determination and SLRP Requantization

Voice Activity and Double-talk information

Speech Level Measurement

Figure 16

Index of x_{max} → Q^{-1} → x_{max} → ⊗ → $x_{max, ALC}$ → Q → Index of requantized x_{max}

Desired ALC Gain

Figure 17

At the Encoder

$G(n)$ → $R(n)$ → Q → $R(n)1$

$P_1(z)$

At the Decoder

$R_{alc}(n)$ → $I + P_1(z)$ → $G_{alc}(n)$

At the Coded Domain ALC Device

$\Delta G(n)$ → Q → $\Delta R(n)$

$P_1(z)$

Figure 18

(a) desired gain
(b) Realized gain with quantizer in the
        feedback loop
(c) Realized gain with quantizer outside the
        feedback loop

Figure 19

At the Encoder

At the Decoder

At the Coded Domain ALC Device
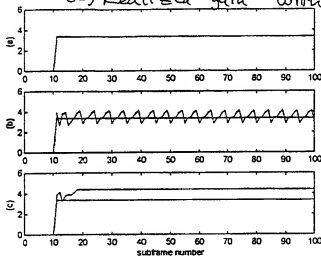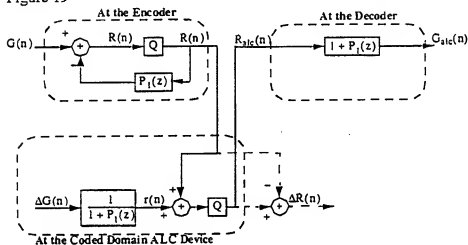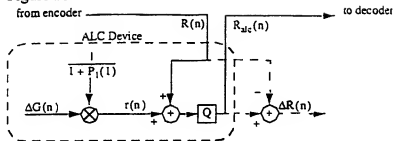
Figure 20
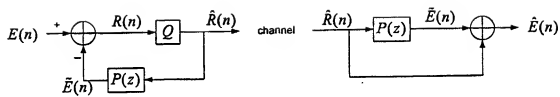

Figure 21

Fig 22

Fig 23

Fig. 24

$E(n)$ ⟶ $+$ ⊕ ⟶ $R(n)$ ⟶ $\boxed{Q}$ ⟶ $\hat{R}(n)$ ⟶ channel ⟶ $\hat{R}(n)$ ⟶ $\boxed{P(z)}$ ⟶ $\tilde{E}(n)$ ⊕ ⟶ $\hat{E}(n)$

$\tilde{E}(n)$ $\boxed{P(z)}$
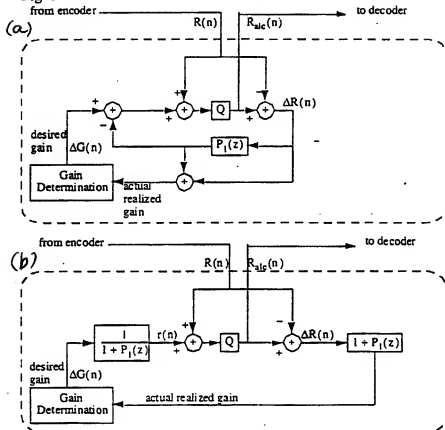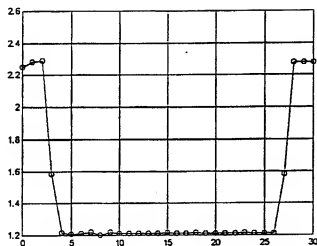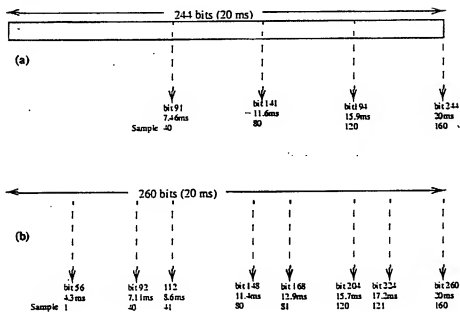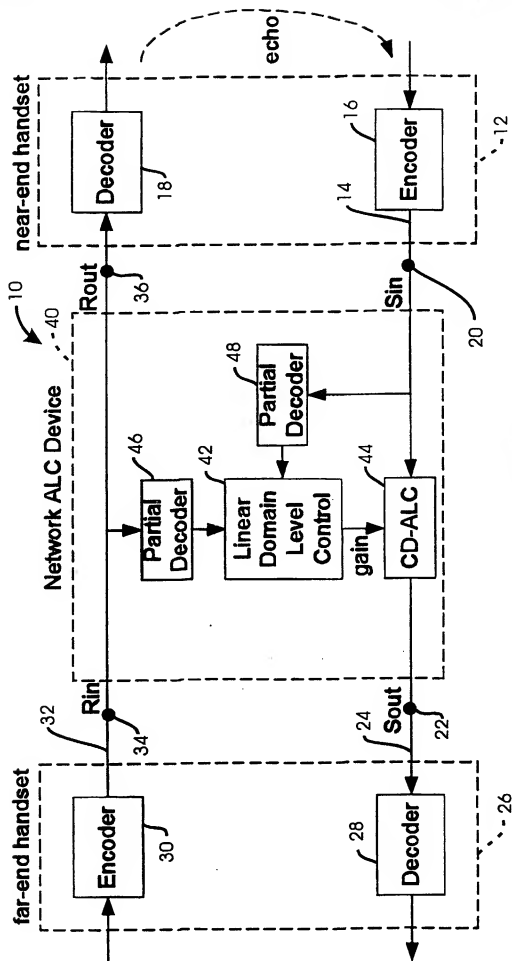
Figure 25

Fig. 26

Fig. 27

Fig. 28
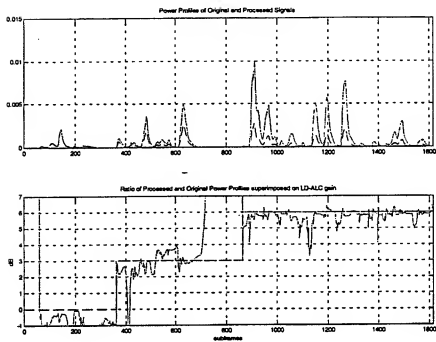
Fig. 29

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04B 1/66

US CL :375/240

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 375/240; 704/205, 207, 211, 260, 270, 278

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST (audio or voice compression, adjust and parameters, decoding)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,847,303 A (MATSUMOTO) 08 DECEMBER 1998, abstract, figures 8 and 13, col. 1, line 50 to col. 3, line 52, and claims 1-22	1-9, 11-18, 24-39, 45-48, 50-52, and 54-60
A	US 5,828,995 A (SATYAMURTI et al) 27 OCTOBER 1998, abstract, figure 4, col. 1, line 20 to col. 2, line 4.	1-60
A	US 5,140,543 A (TANAKA) 18 AUGUST 1992, abstract, col 1, line 6 to col. 2, line 3, and claims 1-11	1-60

☐ Further documents are listed in the continuation of Box C.   ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
12 SEPTEMBER 2000	1 0 OCT 2000

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	MOHAMMAD GHAYOUR
Facsimile No. (703) 305-3230	Telephone No. (703) 306-3034